Fast campus

2017 Machine Learning with R

Clustering

강필성

고려대학교 산업경영공학부

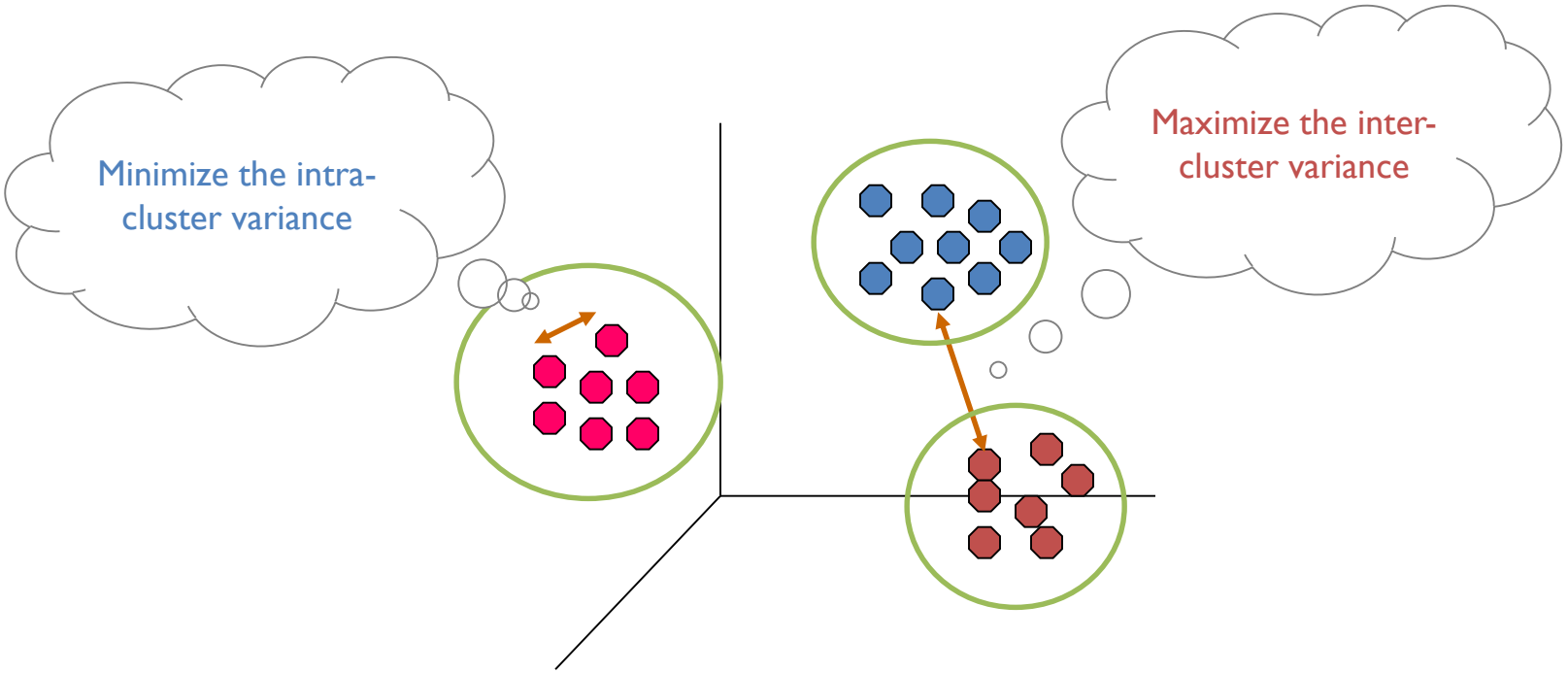
pilsung_kang@korea.ac.kr

목차

군집화: Clustering

❖ 군집화(Clustering)

- 관측치들의 집단을 판별
  - ✓동일한 집단에 소속된 관측치들은 서로 유사할수록 좋음
  - ✓상이한 집단에 소속된 관측치들은 서로 다를수록 좋음

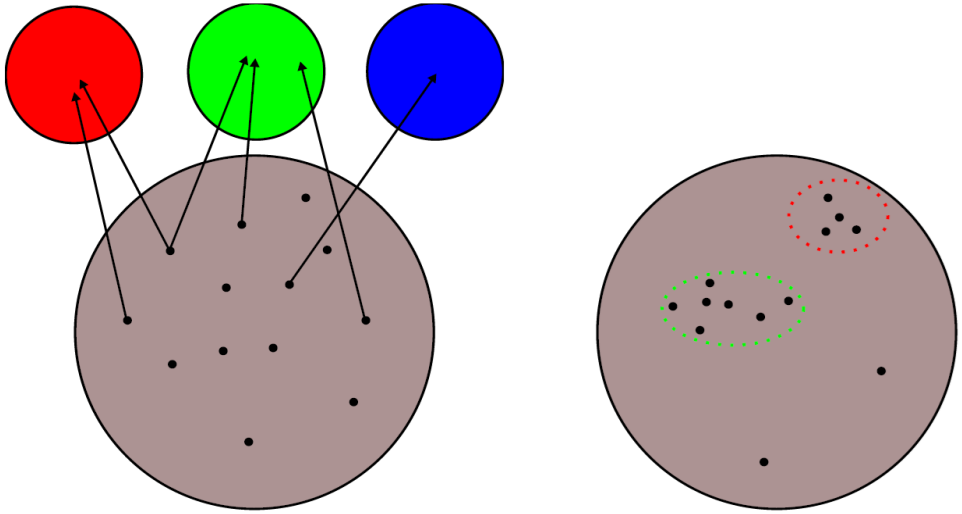


군집화: Clustering

Andrews and Fox (2007)

❖ 분류 (Classification) vs. 군집화(Clustering)

- 분류(Classification): 범주의 수 및 각 개체의 범주 정보를 사전에 알 수 있으며, 개체의 입력 변수 값들로부터 범주 정보를 유추하여 새로운 개체에 대해 가장 적합한 범주로 할당하는 문제 (supervised learning)
- 군집화(Clustering): 군집의 수, 속성, 멤버십 등이 사전에 알려져 있지 않으며 최적의 구분을 찾아가는 문제 (unsupervised learning)



(a) Classification (b) Clustering

군집화: 적용사례

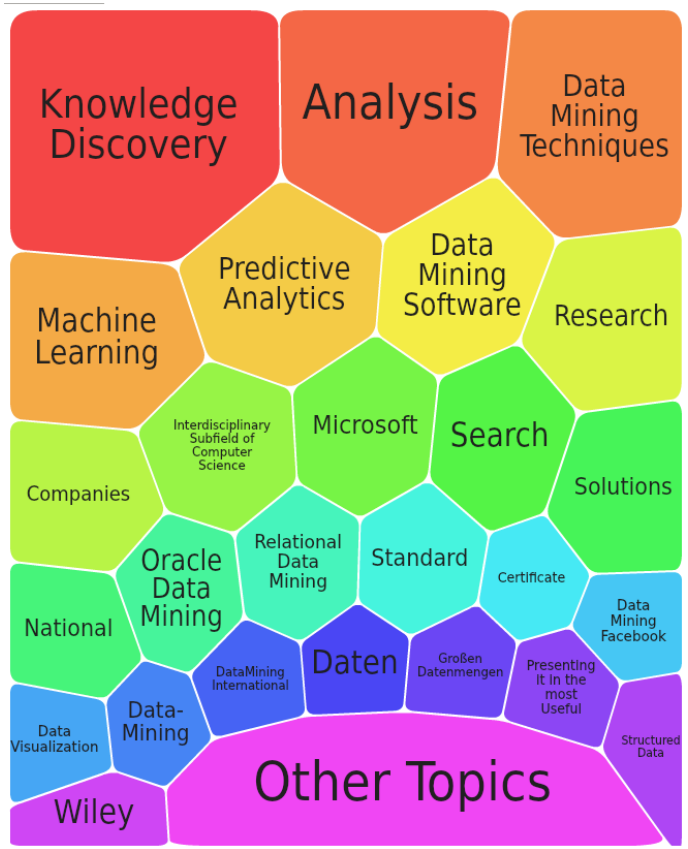
❖ 군집화 적용 사례

▪ 데이터에 대한 이해

- ✓ 웹브라우징 시 유사한 문서들을 표시
- ✓ 유사한 기능을 수행하는 유전자/단백질 집합
- ✓ 유사한 추세를 나타내는 주식 종목들 등

Query: israel
Documents: 272, Clusters: 15, Average Cluster Size: 15.1 documents

Cluster	Size	Shared Phrases and Sample Document Titles
1 View Results Refine Query Based On This Cluster	16	Society and Culture (56%), Faiths and Practices (56%), Judaism (69%), Spirituality (56%); Religion (56%) , organizations (43%) Ahavat Israel - The Amazing Jewish Website! Israel and Judaism Judaica Collection
2 View Results Refine Query Based On This Cluster	15	Ministry of Foreign Affairs (33%), Ministry (87%) Publications and Data of the BANK OF ISRAEL Consulate General of Israel to the Mid-Atlantic Region The Friends of Israel Gospel Ministry
3 View Results Refine Query Based On This Cluster	11	Israel Tourism (36%), Comprehensive Israel (36%), Tourism (64%) Interactive Israel tourism guide - Jerusalem Ambassade d'Israel Travel to Israel Opportunites
4 View Results Refine Query Based On This Cluster	7	Middle East (57%), History (57%); WAR (42%) , Region (42%) , Complete (42%) , Listing (42%) , country (42%) Israel at Fifty: Our Introduction to The Six Day War Machal - Volunteers in the Israel's War of Independence HISTORY: The State of Israel
5 View Results Refine Query Based On This Cluster	22	Economy (68%), Companies (55%), Travel (55%) Israel Hotel Association Israel Association of Electronics Industries Focus Capital Group - Israel

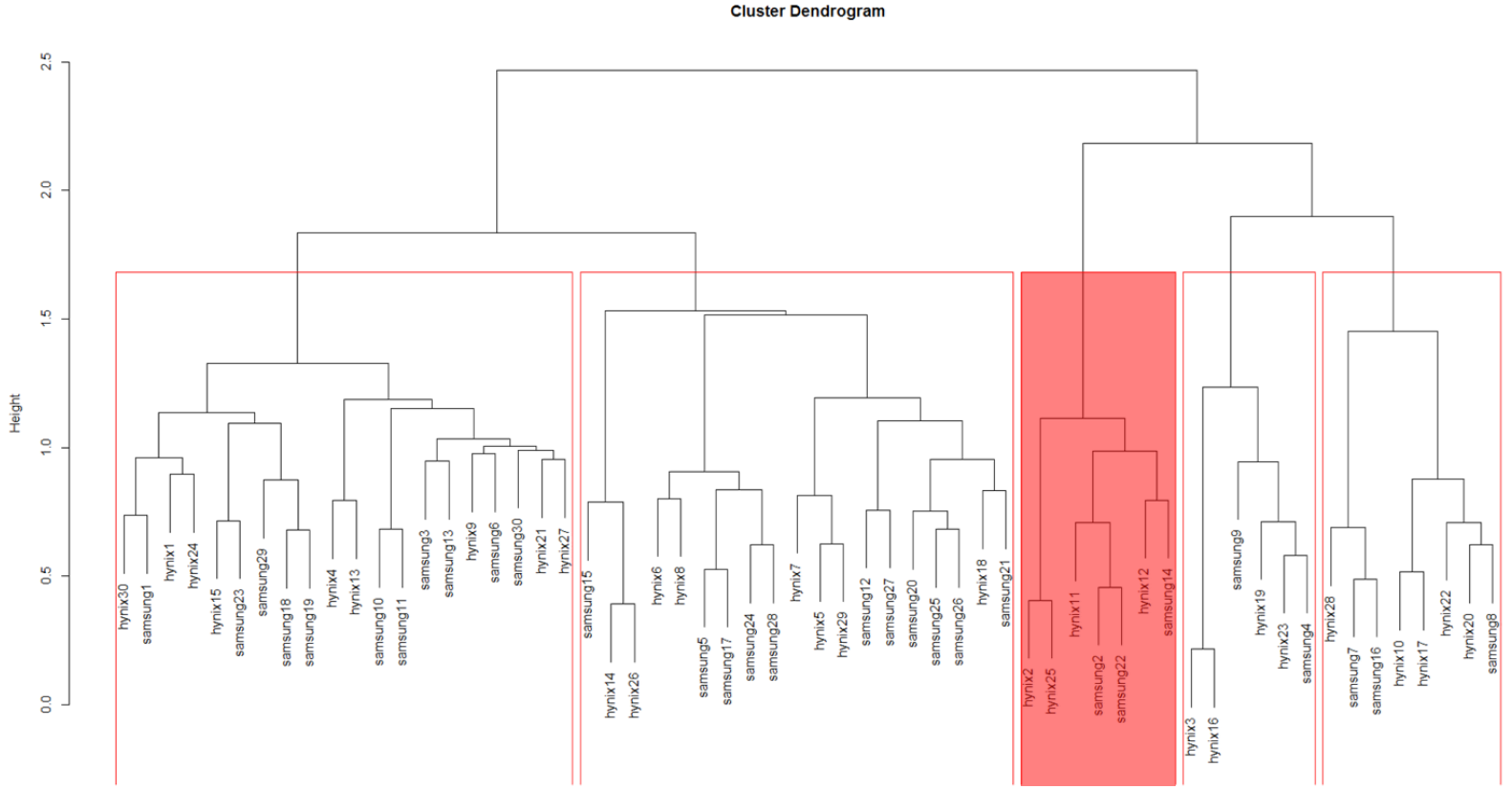


군집화: 적용사례

❖ 군집화 적용 사례

▪ 전략 수립

✓ 경쟁사와의 특허 문서 분석을 통한 장단점 파악



군집화: 적용사례

❖ 군집화 적용 사례

▪ 전략 수립

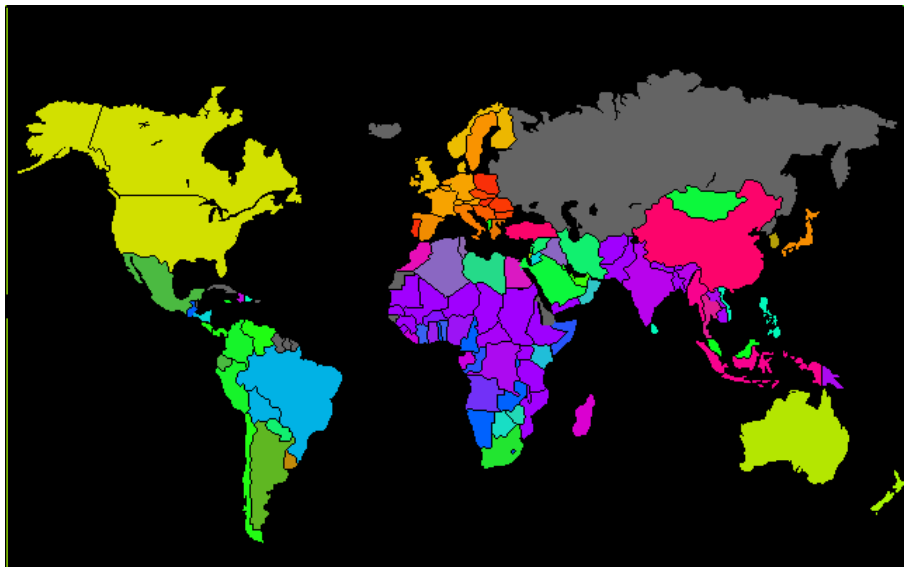
✓ 경쟁사와의 특허 문서 분석을 통한 장단점 파악

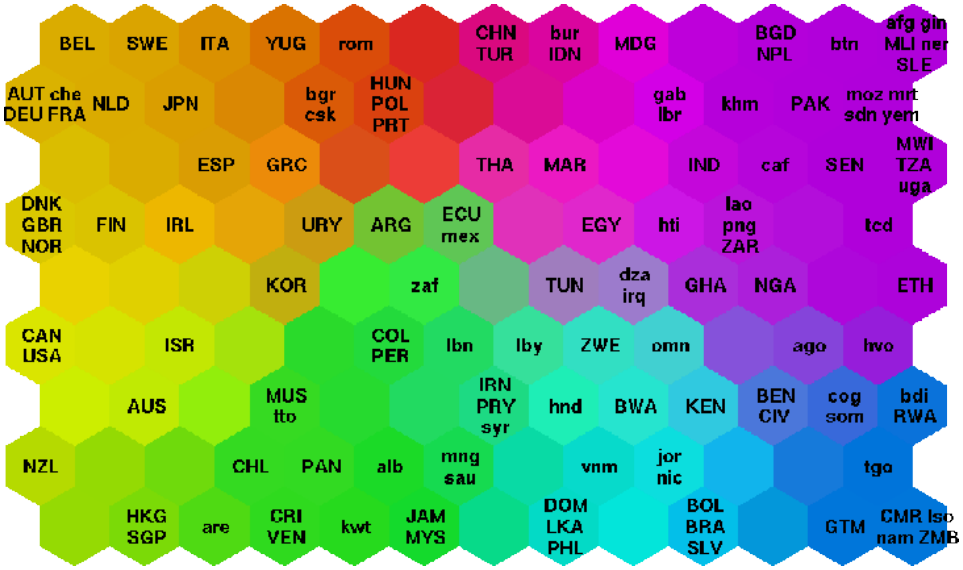
	회사	일련번호	특허명	초록
2	SK하이닉스	2	멀티 레귤레이터 회로 및 이를 구비한 집적회로	본 기술에 따른 레귤레이터 회로는, 입력전압을 일정한 전압 레벨로 레귤레이팅하여 출력하도록 구성된 레귤레이터 및 복수개의 전압 생성 코드 들에 의해 결정되는 내부 저항값들에 따라 상기 레귤레이터의 출력 전압을 분배한 분배전압들을 각각 출력하도록 구성된 복수개의 전압 분배회로를 포함한다.
3	SK하이닉스	11	내부 전압 생성 회로 및 그의 동작 방법	펌핑 동작을 통해 내부 전압을 생성하는 내부 전압 생성 회로에 관한 것으로, 다수의 펌핑부를 포함하며, 목표 전압 레벨에 대응하는 최종 펌핑 전압을 생성하기 위한 펌핑 전압 생성부, 및 상기 목표 전압 레벨에 대응하여 상기 다수의 펌핑부의 활성화 개수를 제어하기 위한 활성화 제어부를 구비하는 내부 전압 생성 회로가 제공된다.
4	SK하이닉스	12	자기 메모리 장치를 위한 라이트 드라이버 회로 및 자기 메모리 장치	비트라인과 소스라인 간에 접속되며, 비트라인 방향으로 인접하는 한 쌍의 자기 메모리 셀이 소스라인을 공유하는 복수의 자기 메모리 셀로 이루어진 메모리 셀 어레이를 포함하는 자기 메모리 장치를 위한 라이트 드라이버 회로로서, 정의 기록전압 공급단자와 부의 기록전압 공급단자 간에 접속되어, 라이트 인에이블 신호 및 데이터 신호에 따라 정의 기록전압 또는 부의 기록전압에 의한 전류를 비트라인에 선택적으로 공급하는 스위칭부를 포함하는 자기 메모리 장치를 제공한다.
5	SK하이닉스	25	전압 레귤레이터 및 전압 레귤레이팅 방법	전압 레귤레이터는 출력전압을 전압 출력단으로 출력하는 전압 출력부와, 제1 제어코드의 제어에 따라 분배 저항값을 조절하는 제1 저항분배 스테이지와, 제1 저항분배 스테이지에서 결정된 분배 저항값을 제2 제어코드의 제어에 따라 조절하는 제2 저항분배 스테이지;를 포함하며, 전압 출력단을 통해서 출력되는 출력전압의 전압레벨은 제1 및 제2 저항분배 스테이지를 통해서 결정된 상기 분배 저항값과, 기준저항의 저항값 비율에 따라 조절되는 것을 특징으로 한다.
6	삼성전자	2	전압 공급 장치 및 그것을 포함한 불휘발성 메모리 장치	본 발명에 따른 전압 공급 장치는 전원 전압을 승압하고, 상기 승압된 전압을 출력 라인으로 제공하기 위한 전하 펌프 및 상기 출력 라인의 전압 레벨을 목표 전압 레벨로 유지하기 위한 전압 제어 회로를 포함한다. 본 발명에 따른 상기 전압 제어 회로는 웰 상에 형성된 제 1 영역 및 제 2 영역을 포함하고, 상기 제 1 영역 및 제 2 영역 사이의 리치 스루(reach through)를 이용하여 상기 출력 라인의 전압 레벨을 제어하기 위한 리치 스루 소자를 포함한다.
7	삼성전자	14	파워 공급 회로 및 이를 구비하는 상 변화 메모리 장치	파워 공급 회로 및 이를 구비하는 상 변화 메모리 장치가 개시된다. 본 발명의 제 1 실시예에 따른 반도체 메모리 장치는 파워 공급 회로, 스위치들 및 선택기들을 구비한다. 파워 공급 회로는 상기 블록들의 메모리 셀들에 사용되는 제 1 전압 및 제 2 전압을 생성한다. 스위치들은 상기 파워 공급 회로와 상기 제 1 전압이 전달되는 제 1 라인 및 상기 제 2 전압이 전달되는 제 2 라인으로 연결되고, 제어 신호에 응답하여 상기 제 1 전압 및 제 2 전압 중 하나를 대응되는 블록으로 인가한다. 선택기들은 블록 선택 신호 및 디스차아지 성공 신호에 응답하여, 상기 제어 신호를 생성한다. 본 발명에 따른 파워 공급 회로 및 이를 구비하는 상 변화 메모리 장치는 셀 블록마다 별도의 파워 스위치를 구비함으로써 파워 공급 회로의 동작 시간 및 동작 전류를 감소시킬 수 있다. 또한, 기입 전압을 디스차아지한 후 다른 레벨의 전압을 공급함으로써, 상 변화 메모리 장치의 오작동이 방지될 수 있다.
8	삼성전자	22	전압 안정화 장치 및 그것을 포함하는 반도체 장치 및 전압 생성 방법	본 발명은 전압 안정화 장치 및 그것을 이용하는 반도체 장치에 관한 것이다. 본 발명의 기술적 사상의 실시예에 따른 전압 안정화 장치는 제 1 전압을 생성하는 제 1 레귤레이터 및 상기 제 1 전압보다 낮은 제 2 전압을 생성하는 제 2 레귤레이터를 포함하되, 상기 제 2 레귤레이터는 상기 제 1 전압의 레벨과 미리 정해진 기준 전압의 레벨의 비교 결과에 기초하여 상기 제 1 전압 또는 상기 제 1 전압보다 높은 제 3 전압을 선택적으로 이용하여 상기 제 2 전압을 생성한다. 본 발명의 기술적 사상의 실시 예에 따르면 제 1의 전압≥제 2의 전압의 관계를 유지하면서, 동시에 제 2의 전압을 고속으로 전위 변환 시킬 수 있다.

군집화: 적용사례

❖ 군집화 적용 사례

- 대량의 데이터에 대한 요약
  - ✓고차원의 데이터를 저차원으로 축약하여 정보를 요약
- 시각화(Visualization)과 밀접한 관계가 있음

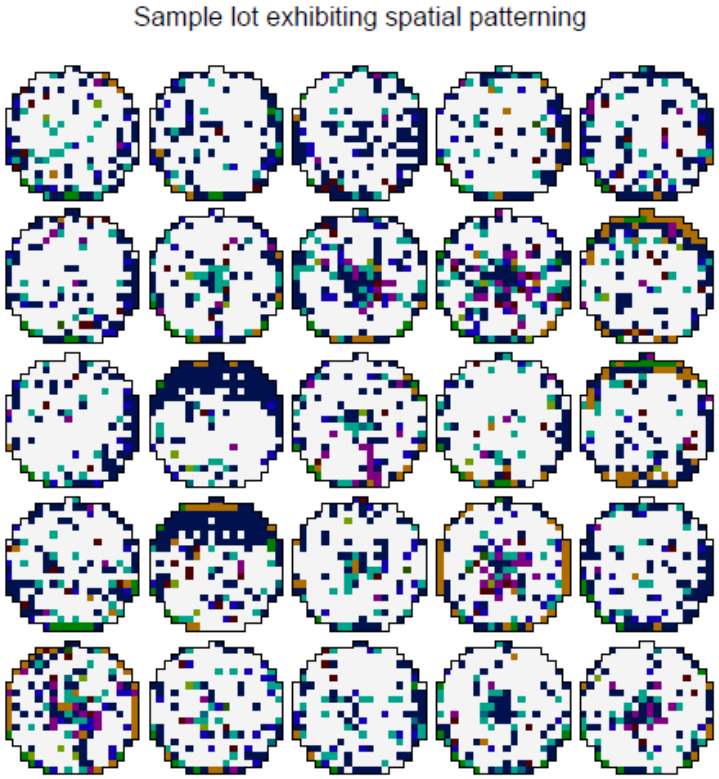


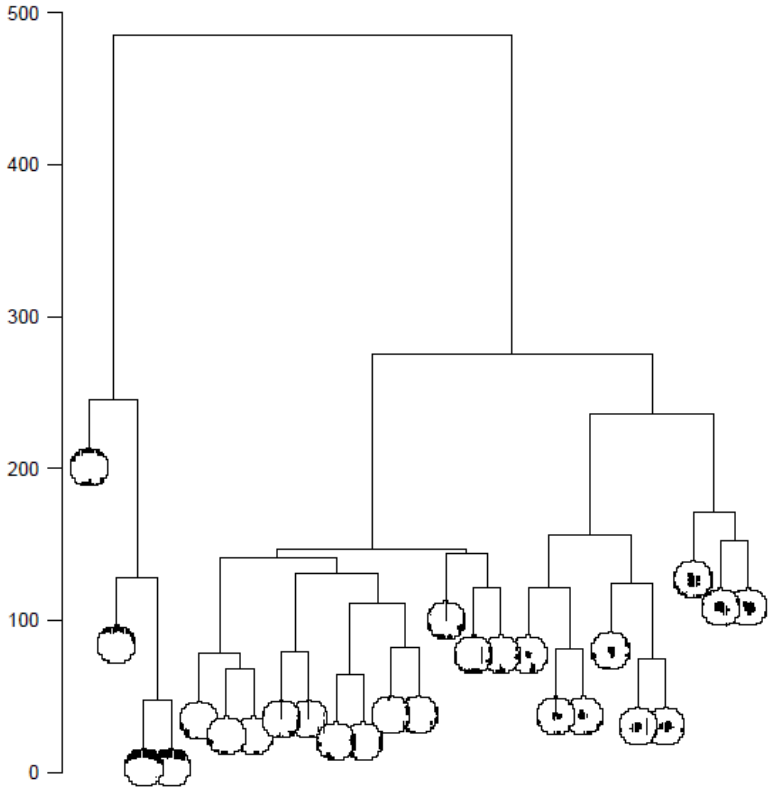


군집화: 적용사례

❖ 군집화 적용 사례

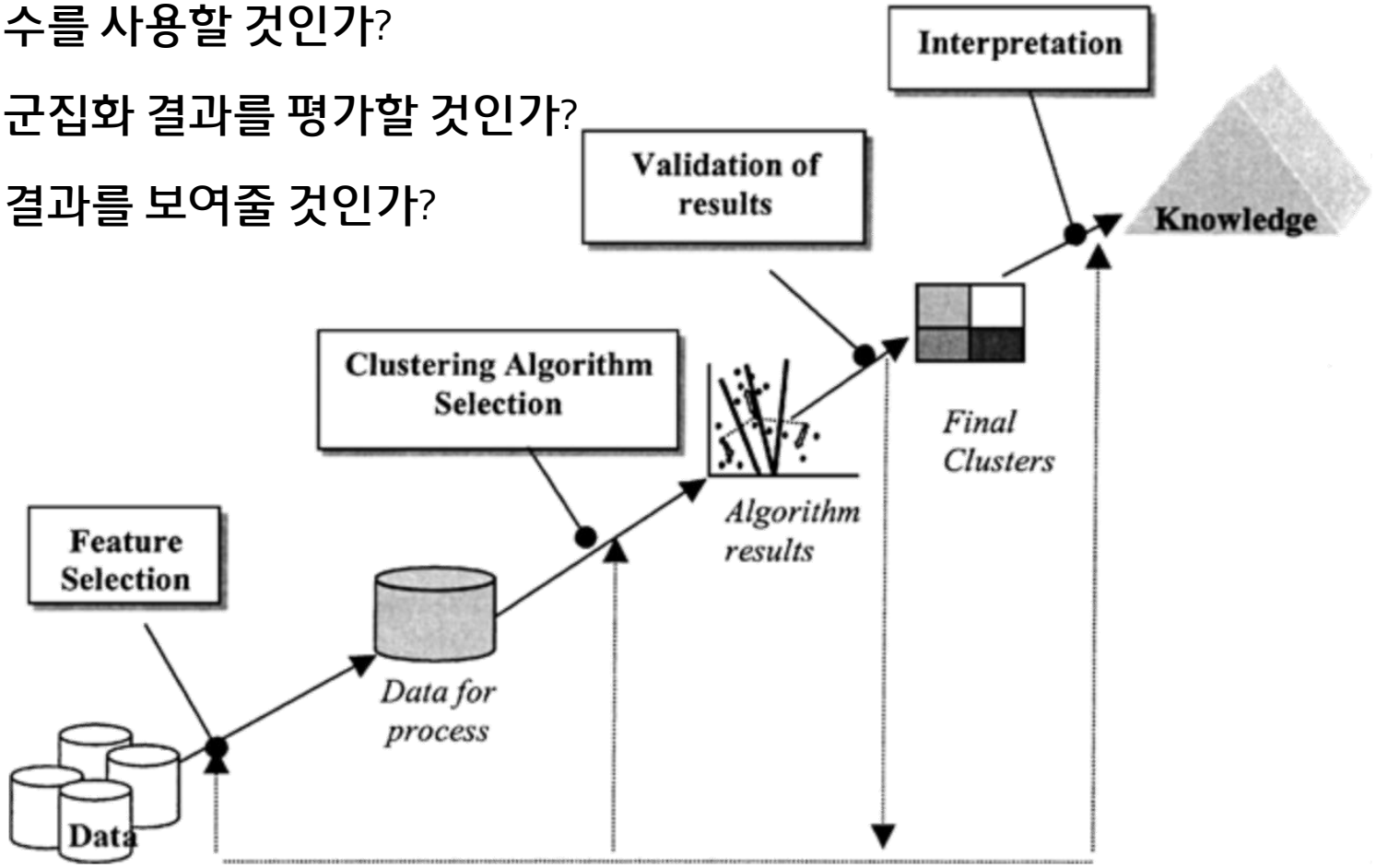
- 웨이퍼 Fail bit map 군집화





군집화: 수행 절차

- 일반적인 군집화 수행 절차
  - 어떤 변수를 사용할 것인가?
  - 어떻게 군집화 결과를 평가할 것인가?
  - 어떻게 결과를 보여줄 것인가?



http://web.itu.edu.tr/sgunduz/courses/verimaden/paper/validity_survey.pdf

군집화: 고려 사항

❖ 군집화 수행 시 고려 사항

- **최적의 군집 수 결정**

How many clusters?

Six Clusters

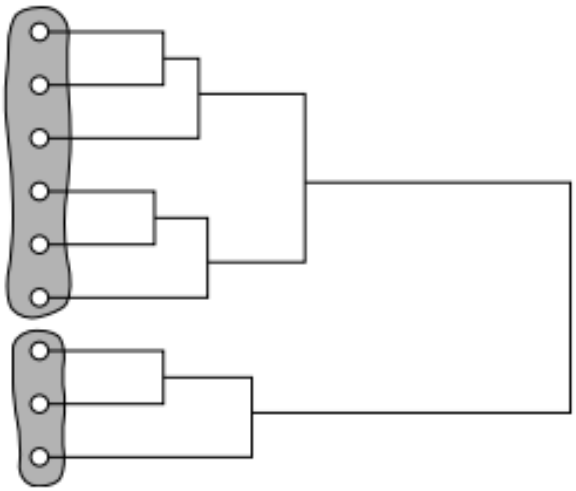
Two Clusters

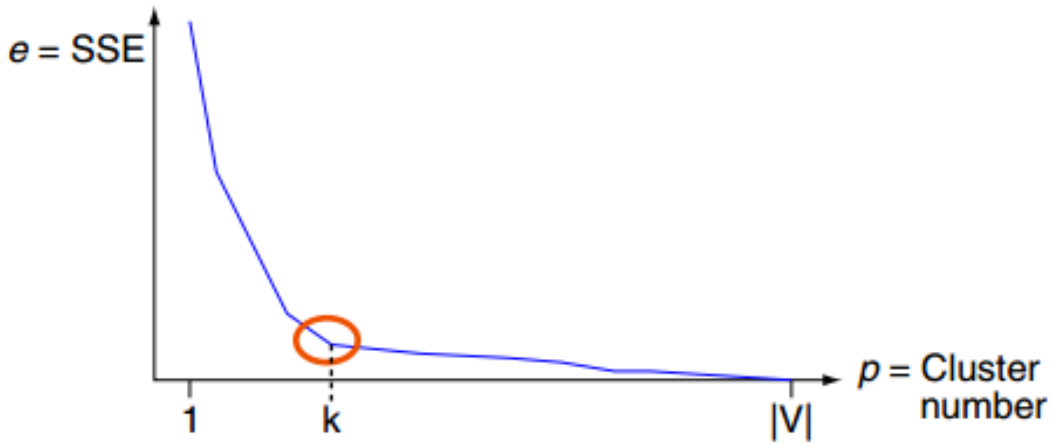
Four Clusters

군집화: 고려 사항

❖ 군집화 수행 시 고려 사항

- 최적의 군집 수 결정
  - ✓ 다양한 군집 수에 대해 성능 평가 지표를 도시하여 최적의 군집 수 선택
  - ✓ Elbow point에서 최적 군집 수가 결정되는 경우가 일반적임





군집화: 고려 사항

- 군집화 수행 시 고려 사항
  - 군집화 결과를 어떻게 평가할 것인가?
  - 분류/회귀 알고리즘처럼 모든 상황에서 적용가능한 Global Performance Measure 부재
- 군집화 평가 지표는 다음과 같이 세 가지 카테고리로 구분할 수 있음
  - External: 정답 레이블과의 비교를 통해 성능 평가 (현실적으로 불가능)
  - Internal: "군집이 얼마나 컴팩트한가"에 보다 초점을 둠
  - Relative: "군집이 얼마나 컴팩트한가"와 "군집끼리 얼마나 다른가"를 동시에 고려하고자 함

군집화: 고려 사항

❖ 군집화 수행 시 고려 사항

- 군집화 결과를 어떻게 평가할 것인가?
- 분류/회귀 알고리즘처럼 모든 상황에서 적용가능한 Global Performance Measure 부재

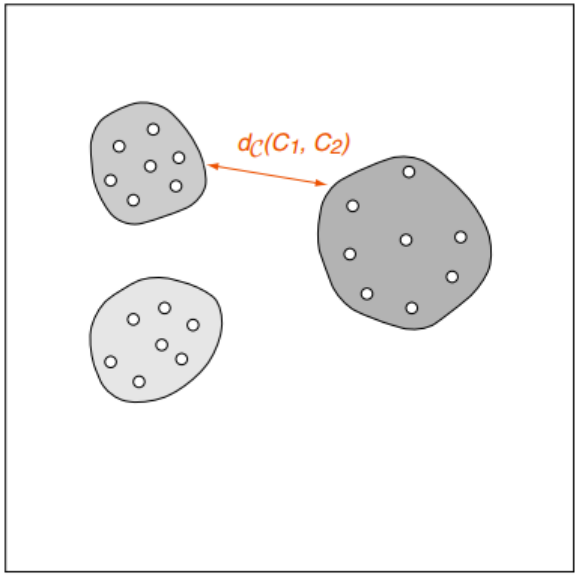
External	Internal	Relative
Rand Statistic	Cophenetic Correlation Coefficient	Dunn family of indices
Jaccard Coefficient	Sum of Squared error (SSE)	Davies-Bouldin (DB) index
Folks and Mallows index	Cohesion and separation	Semi-partial R-squared
(Normalized) Hurbert Γ statistic		SD validity index
		Silhouette

군집화: 고려 사항

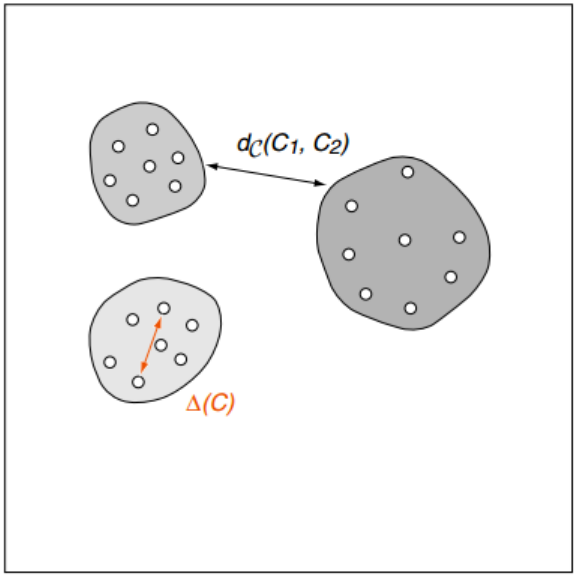
❖ 군집화 평가를 위해 필요한 세 가지 지표 정의

- 군집화가 잘 되어 있다면
  - ✓ (1)번 지표의 값은 크고 (2)번과 (3)번 지표의 값은 상대적으로 작게 나타날 것임

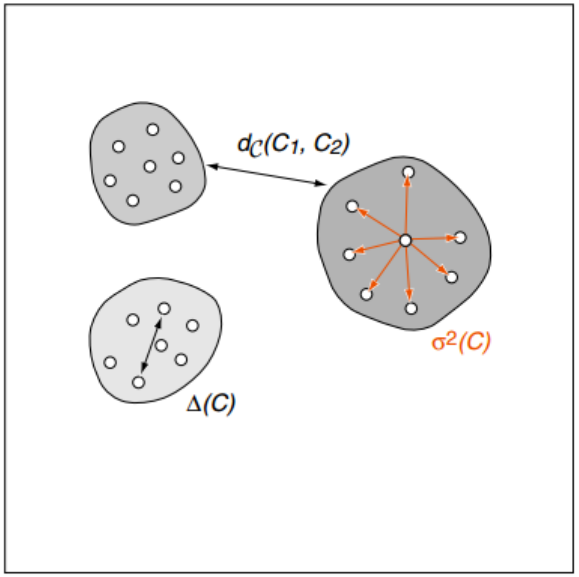
(1) Distance between two clusters



(2) Diameter of a cluster



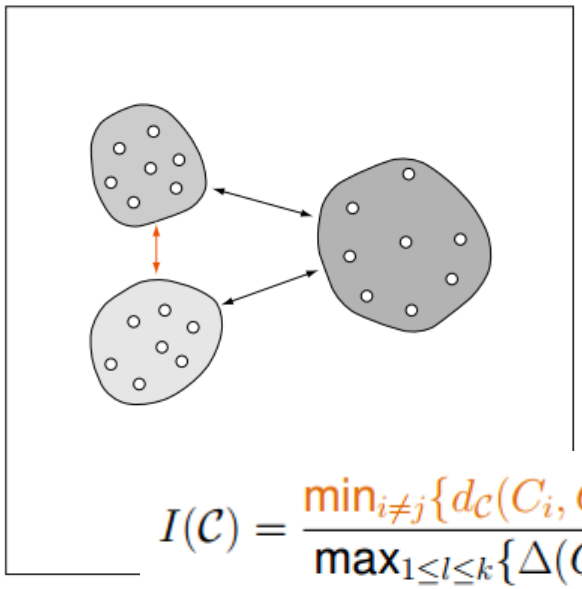
(3) Scatter within a cluster (SSE)



군집화: 고려 사항

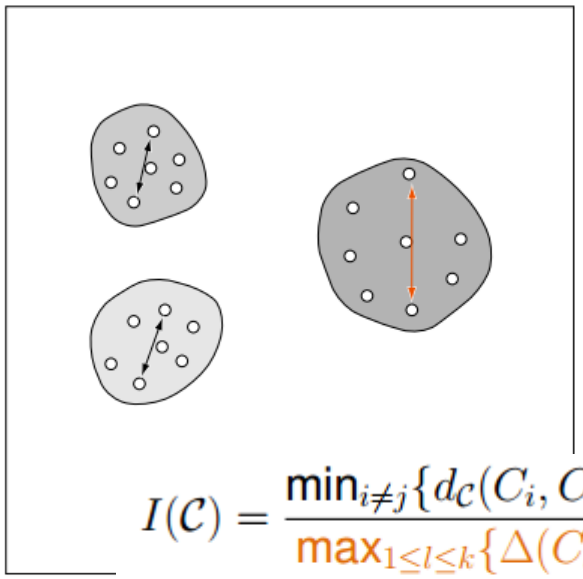
❖ 군집화 평가 지표 1: Dunn Index

- Dunn Index는 군집 내 거리((1)번 지표)중 가장 작은 값을 분자로, 군집의 지름((2)번 지표) 중 가장 큰 값을 분모로 정의함
- Dunn Index는 클수록 우수한 군집화 결과라고 할 수 있음



$$I(\mathcal{C}) = \frac{\min_{i \neq j}\{d_{\mathcal{C}}(C_i, C_j)\}}{\max_{1 \leq l \leq k}\{\Delta(C_l)\}},$$

$$I(\mathcal{C}) \rightarrow \max$$



$$I(\mathcal{C}) = \frac{\min_{i \neq j}\{d_{\mathcal{C}}(C_i, C_j)\}}{\max_{1 \leq l \leq k}\{\Delta(C_l)\}},$$

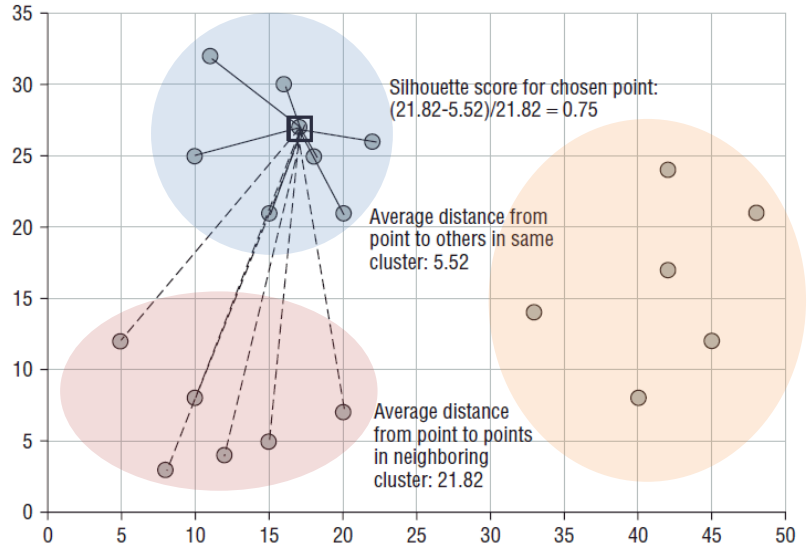
$$I(\mathcal{C}) \rightarrow \max$$

군집화: 고려 사항

❖ 군집화 평가 지표 2: Silhouette

- a(i): 개체 i로부터 같은 군집 내에 있는 모든 다른 개체들 사이의 평균 거리
- b(i): 개체 i로부터 다른 군집 내에 있는 개체들 사이의 평균 거리 중 가장 작은 값

$$s(i) = \frac{b(i) - a(i)}{\max\{a(i), b(i)\}} \qquad s(i) = \begin{cases} 1 - a(i)/b(i), & \text{if } a(i) < b(i) \\ 0, & \text{if } a(i) = b(i) \\ b(i)/a(i) - 1, & \text{if } a(i) > b(i) \end{cases} \qquad -1 \leq s(i) \leq 1$$



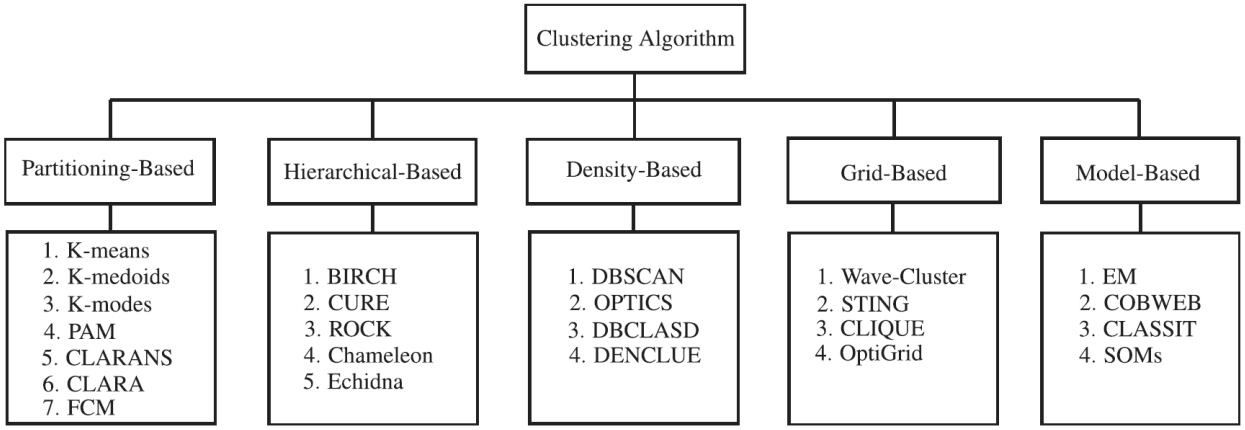
군집화 종류

Fahad et al. (2014)

❖ Hard Clustering vs. Soft Clustering

▪ Hard Clustering (Crisp Clustering)

- ✓서로 겹치지 않는(non-overlapping) 군집 생성
- ✓각 개체는 오직 하나의 군집으로만 할당됨



▪ Soft Clustering (Fuzzy Clustering)

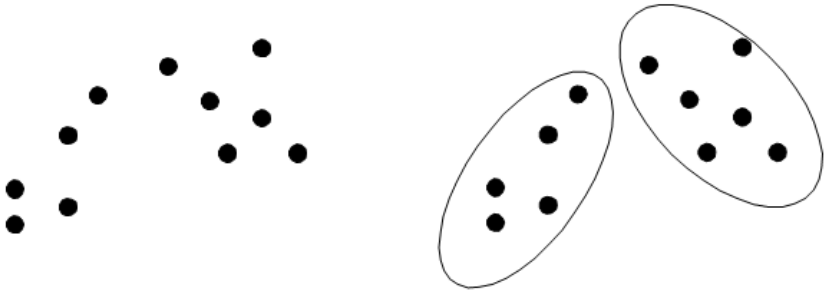
- ✓겹치는 군집을 생성하는 것도 가능함
- ✓한 개체는 여러 개의 군집에 확률적인 할당이 될 수 있음

군집화: 알고리즘

❖ 군집화 알고리즘의 종류

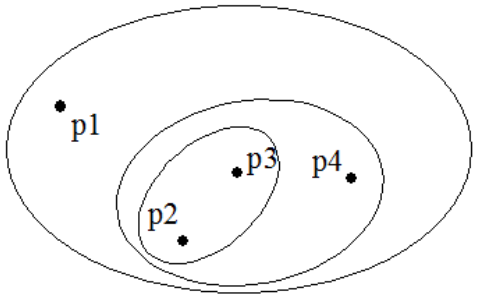
▪ 분리형 군집화

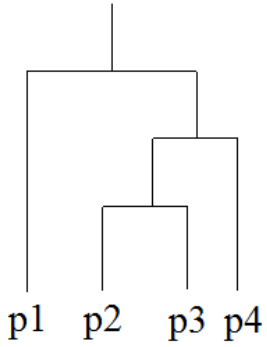
- ✓ 전체 데이터의 영역을 특정 기준에 의해 동시에 구분
- ✓ 각 개체들은 사전에 정의된 군집 수 중 하나에 속하는 결과를 도출함



▪ 계층적 군집화

- ✓ 개체들을 가까운 집단부터 차근차근 묶어가는 방식
- ✓ 군집화 결과 뿐만 아니라 유사한 개체들이 결합되는 절차(dendrogram)도 생성

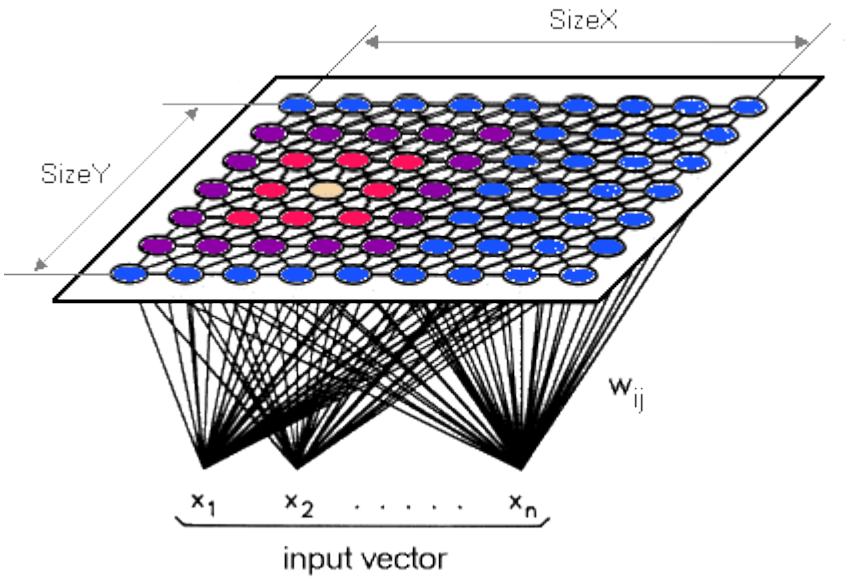




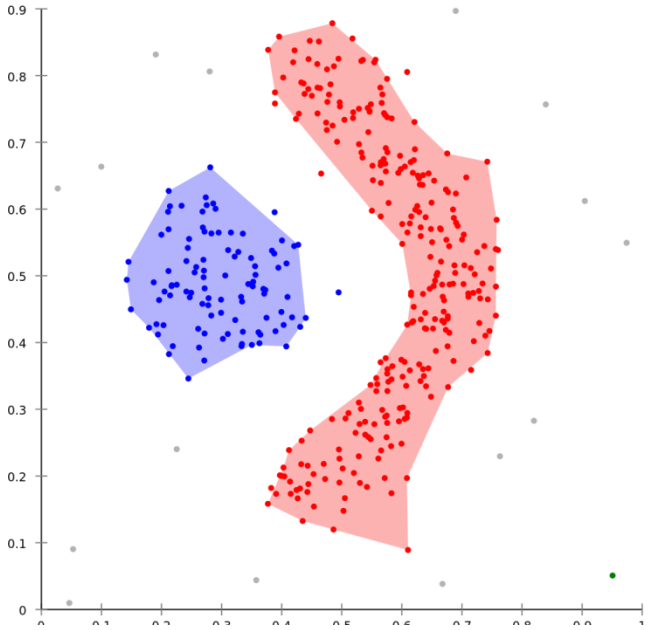
군집화: 알고리즘

❖ 군집화 알고리즘의 종류

- 자기조직화 지도: Self-Organizing Map (SOM)
  - ✓ 2차원의 격자에 각 개체들이 대응하도록 인공신경망과 유사한 학습을 통해 군집 도출



- 밀도 기반 군집화
  - ✓ 데이터의 분포를 기반으로 높은 밀도를 갖는 세부 영역들로 전체 영역을 구분



목차

K-평균 군집화: K-Means Clustering

❖ K-평균 군집화

- 대표적인 분리형 군집화 알고리즘
  - ✓ 각 군집은 하나의 중심(centroid)을 가짐
  - ✓ 각 개체는 가장 가까운 중심에 할당되며, 같은 중심에 할당된 개체들이 모여 하나의 군집을 생성
  - ✓ 사전에 군집의 수 K가 정해저야 알고리즘을 실행할 수 있음

$$\mathbf{X} = C_1 \cup C_2 \ldots \cup C_K, \quad C_i \cap C_j = \phi, \quad i \neq j$$

$$\arg\min_{\mathbf{C}} \sum_{i=1}^{K} \sum_{\mathbf{x}_j \in C_i} ||\mathbf{x}_j - \mathbf{c}_i||^2$$

K-평균 군집화: K-Means Clustering

❖ K-평균 군집화 (K-Means Clustering) 수행 절차

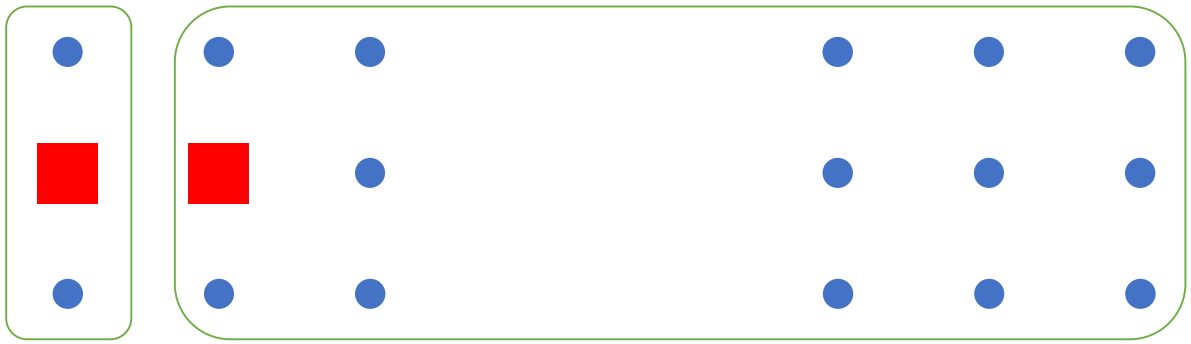
- 1단계: K개의 초기 군집 중심(initial centroid) 설정
- 2단계: 다음 절차를 반복
  - ✓모든 개체를 가장 가까운 군집 중심에 할당하여 군집 구성
  - ✓할당된 개체들을 이용하여 군집 중심을 재설정
  - ✓종료 조건: 모든 군집 중심의 위치가 변하지 않고, 모든 개체의 군집 할당 결과에 변화가 없을 때 알고리즘 종료
- Note: 초기 중심은 종종 무작위로 선택되며 따라서 군집화의 결과가 초기 중심 설정에 따라 다르게 나타나는 경우가 발생할 수도 있음

K-평균 군집화: K-Means Clustering

❖ K-평균 군집화 (K-Means Clustering) 수행 절차 예시

- 1단계: K개의 초기 군집 중심(initial centroid) 설정

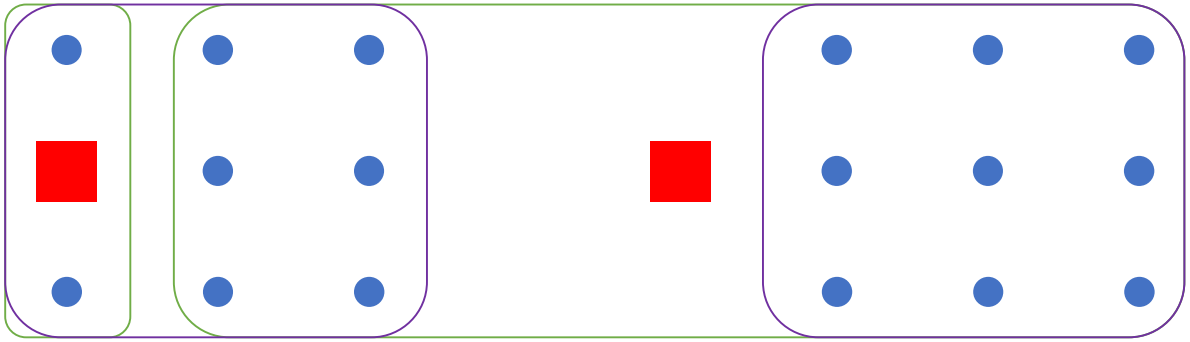
- 2-1단계(1회차): 모든 개체를 가장 가까운 중심에 할당
- 2-2단계(2회차): 할당된 개체들을 이용하여 군집 중심 재설정



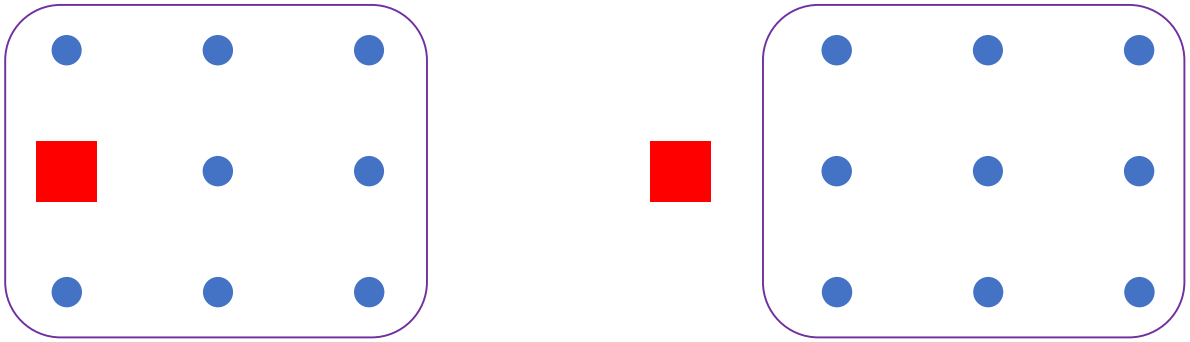
K-평균 군집화: K-Means Clustering

❖ K-평균 군집화 (K-Means Clustering) 수행 절차 예시

- 2-1단계(2회차): 모든 개체를 가장 가까운 중심에 할당



- 2-2단계(2회차): 할당된 개체들을 이용하여 군집 중심 재설정

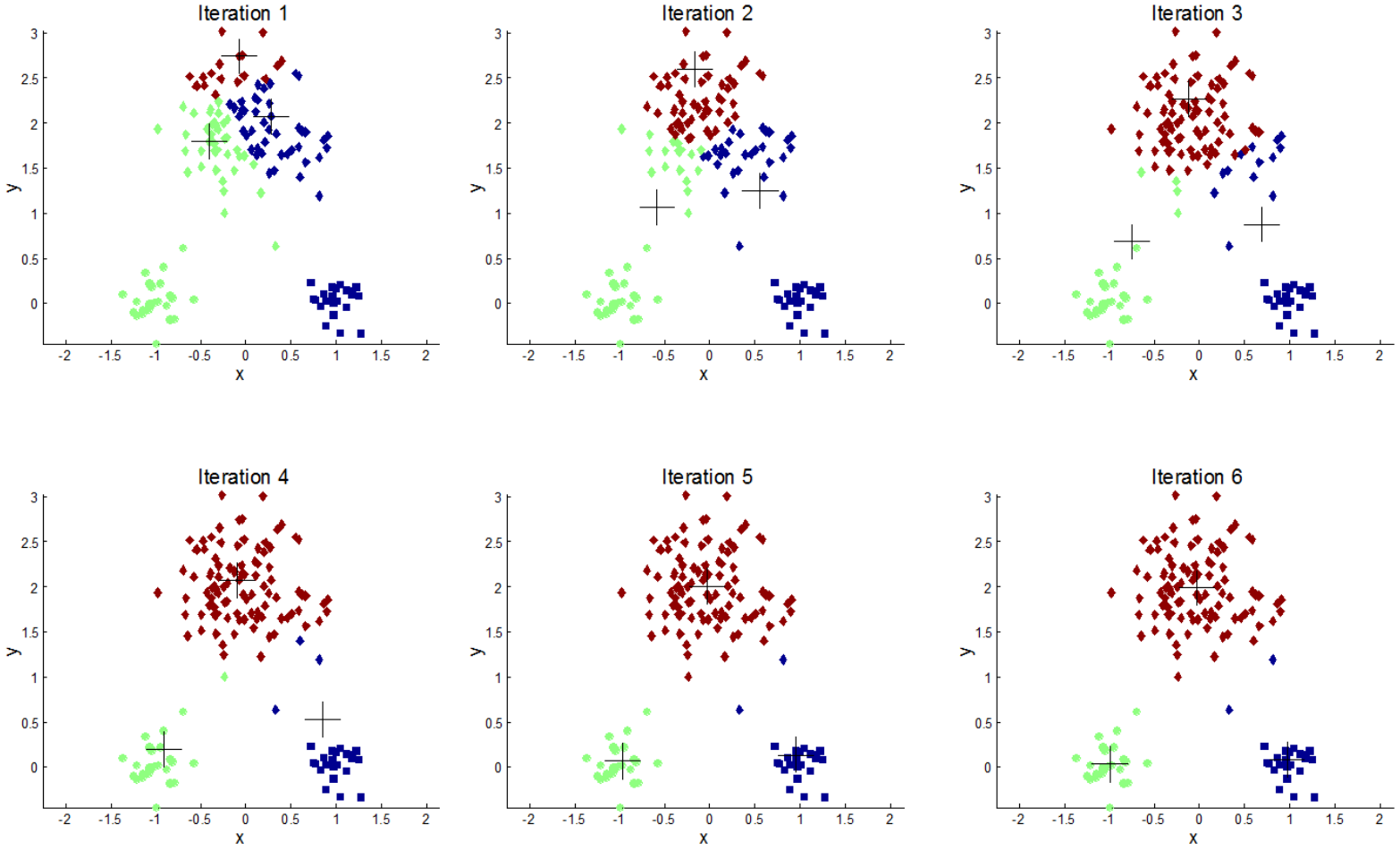


- 군집 중심과 개체 할당에 변화가 없으므로 알고리즘 종료

K-평균 군집화: K-Means Clustering

❖ 초기 중심 설정의 영향

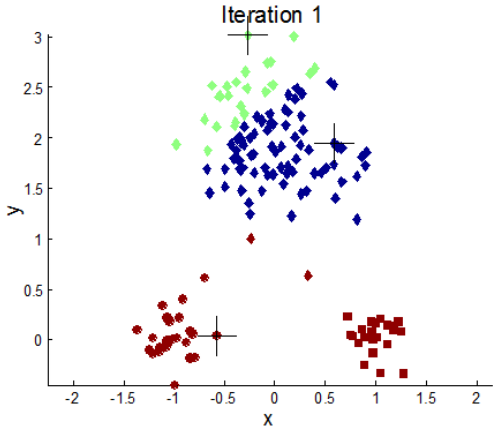
- 바람직한 결과

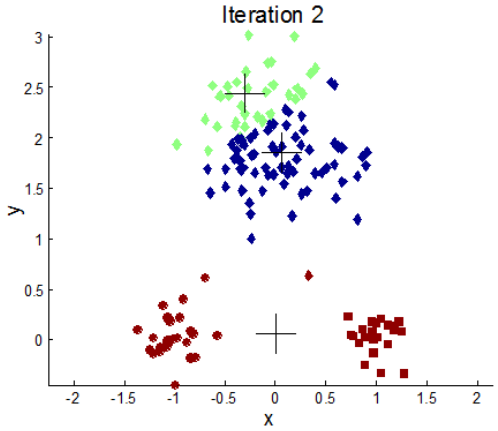


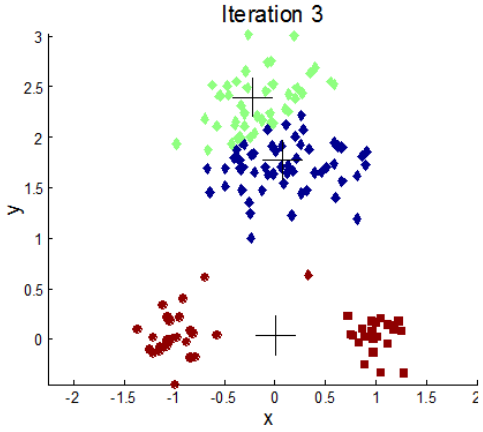
K-평균 군집화: K-Means Clustering

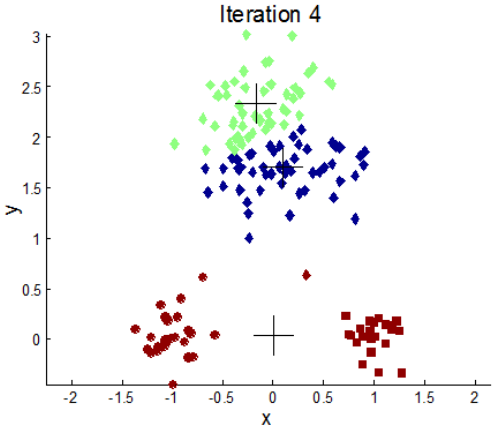
❖ 초기 중심 설정의 영향

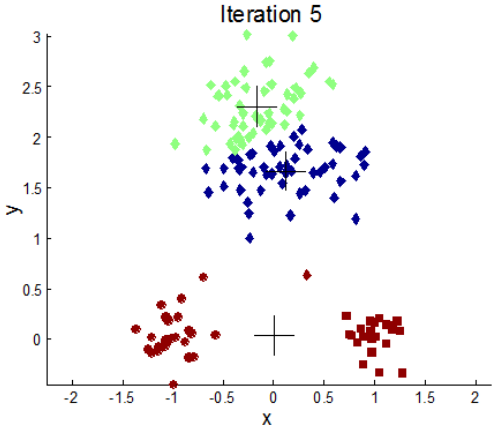
- 바람직하지 않은 결과





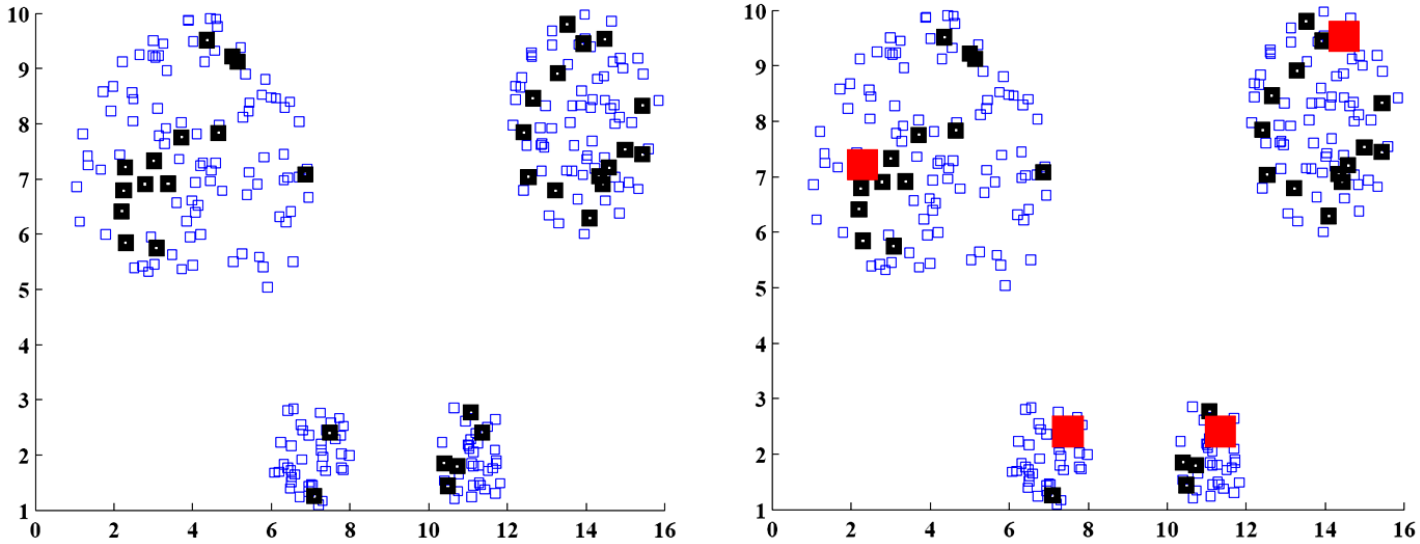






K-평균 군집화: K-Means Clustering

- 무작위 초기 중심 설정의 위험을 감소하고자 하는 다양한 연구가 존재
  - 반복적으로 수행하여 가장 여러 번 나타나는 군집을 이용
  - 전체 데이터 중 일부만 샘플링하여 계층적 군집화를 수행한 뒤 초기 군집 중심 설정
  - 데이터 분포 $\mathcal{L}(\mathbf{x}_s|\mathbf{S},\mathbf{C}) = d_G(\mathbf{x}_s,\mathbf{S}) \times \frac{1}{1+\exp\left(-d_R(\mathbf{x}_s,\mathbf{S})\right)}$

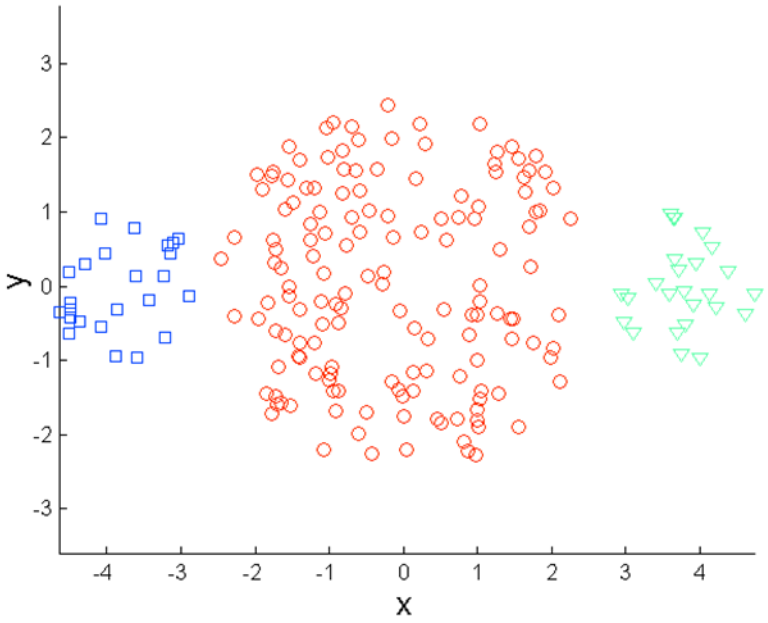


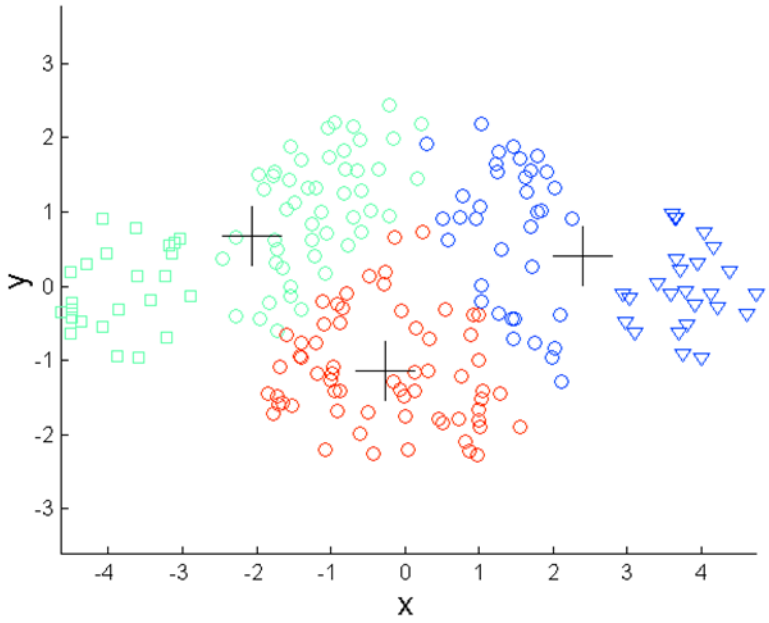
Pilsung Kang and Sungzoon Cho. (2009). K-Means clustering seeds initialization based on centrality, sparsity, and isotropy. *The 13th International Conference on Intelligent Data Engineering and Automated Learning (IDEAL 2009)*, Burgos, Spain. E. Corchado and H. Yin (Eds.), ***Lecture Notes in Computer Science LNCS 5788***, 109-117.

K-평균 군집화: K-Means Clustering

❖ K-평균 군집화의 문제점

- 문제점 1: **서로 다른 크기**의 군집을 잘 찾아내지 못함

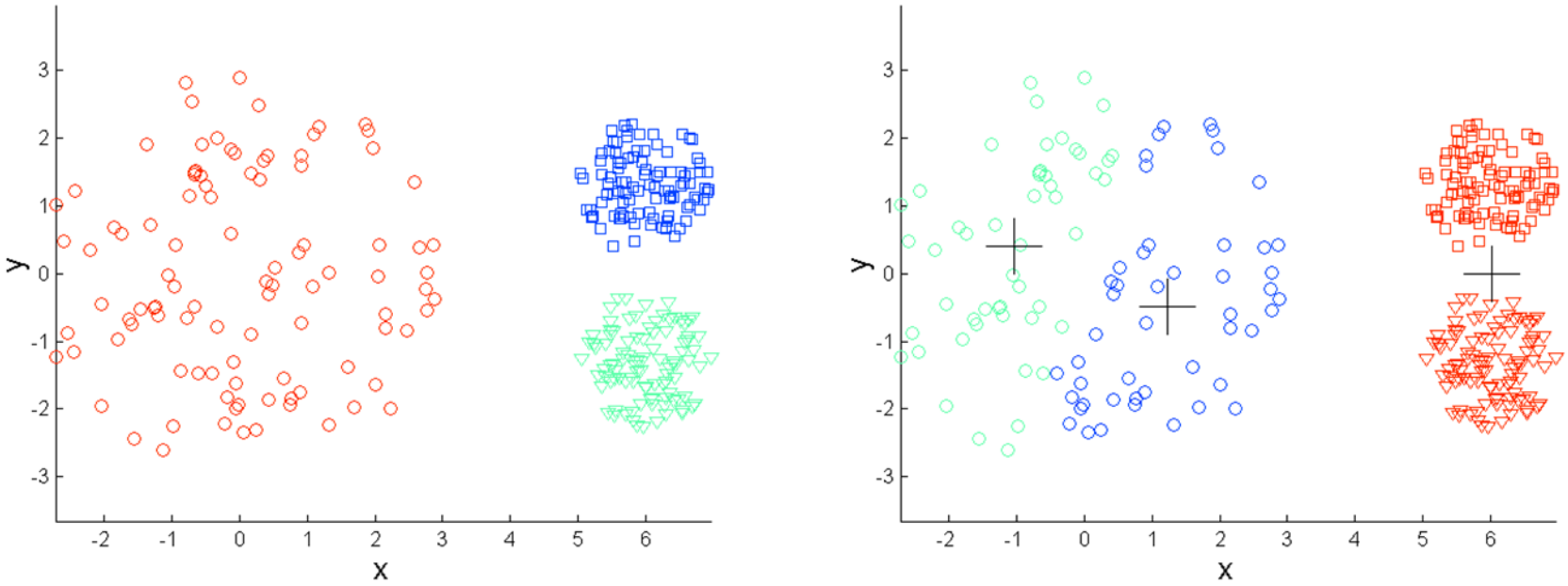




K-평균 군집화: K-Means Clustering

❖ K-평균 군집화의 문제점

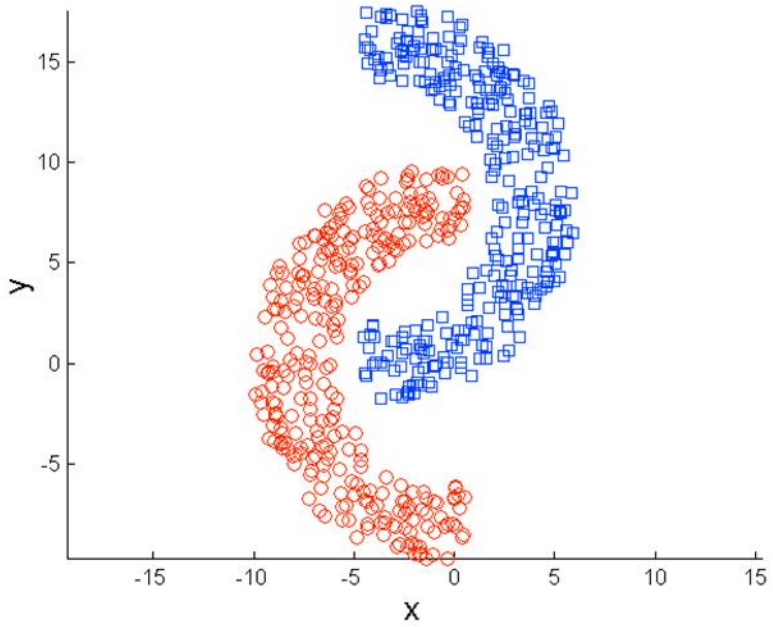
- 문제점 2: **서로 다른 밀도**의 군집을 잘 찾아내지 못함

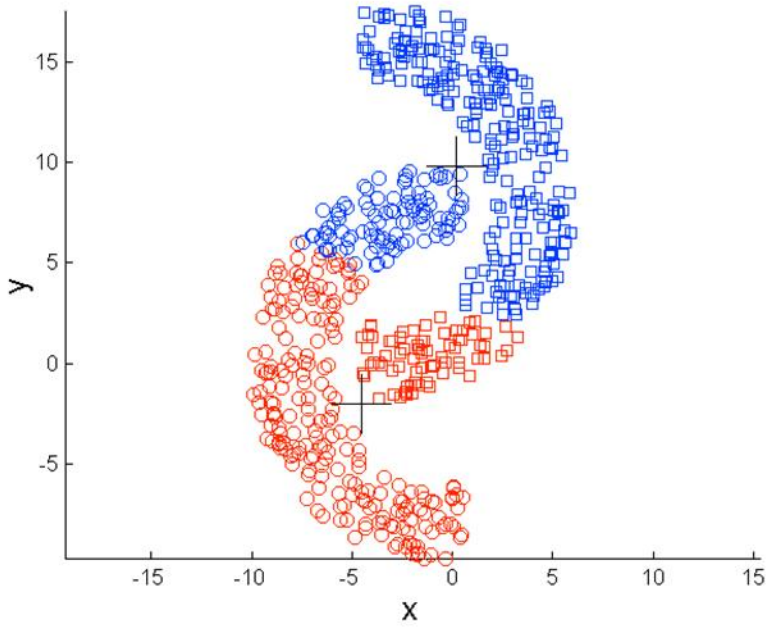


K-평균 군집화: K-Means Clustering

❖ K-평균 군집화의 문제점

- 문제점 3: 구형이 아닌 형태의 군집을 판별하기 어려움



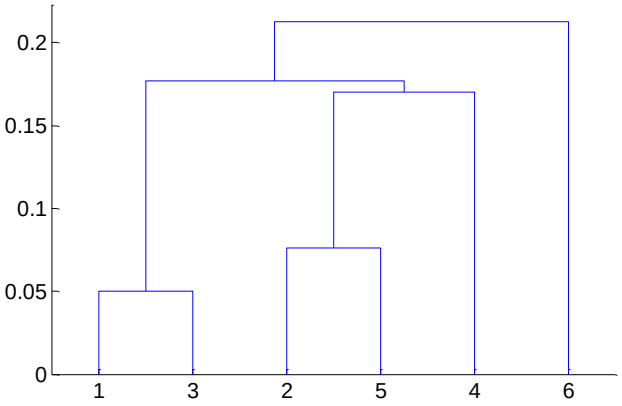


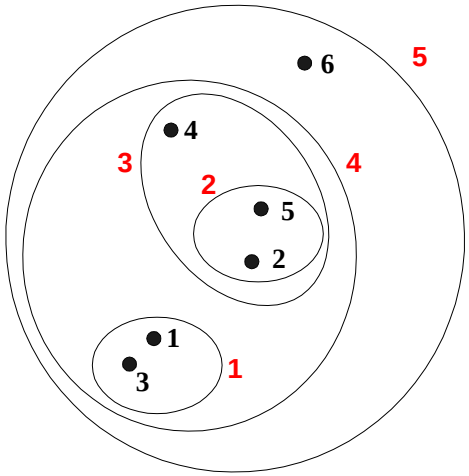
목차

계층적 군집화: Hierarchical Clustering

❖ 계층적 군집화

- 계층적 트리모형을 이용하여 개별 개체들을 순차적/계층적으로 유사한 개체/군집과 통합
- 덴드로그램(Dentrogram)을 통해 시각화 가능
  - ✓Dendrogram: 개체/군집들이 결합되는 순서를 나타내는 트리형태의 구조





계층적 군집화: Hierarchical Clustering

❖ 계층적 군집화의 장점

- 사전에 군집의 수를 정하지 않아도 수행 가능
  - ✓Dendrogram이 생성된 후 적절한 수준에서 자르면 그에 해당하는 군집화 결과 생성
- 특정 분야(domain)에서는 이 dendrogram이 유의미한 계통체계(taxonomies)를 표현하기도 함

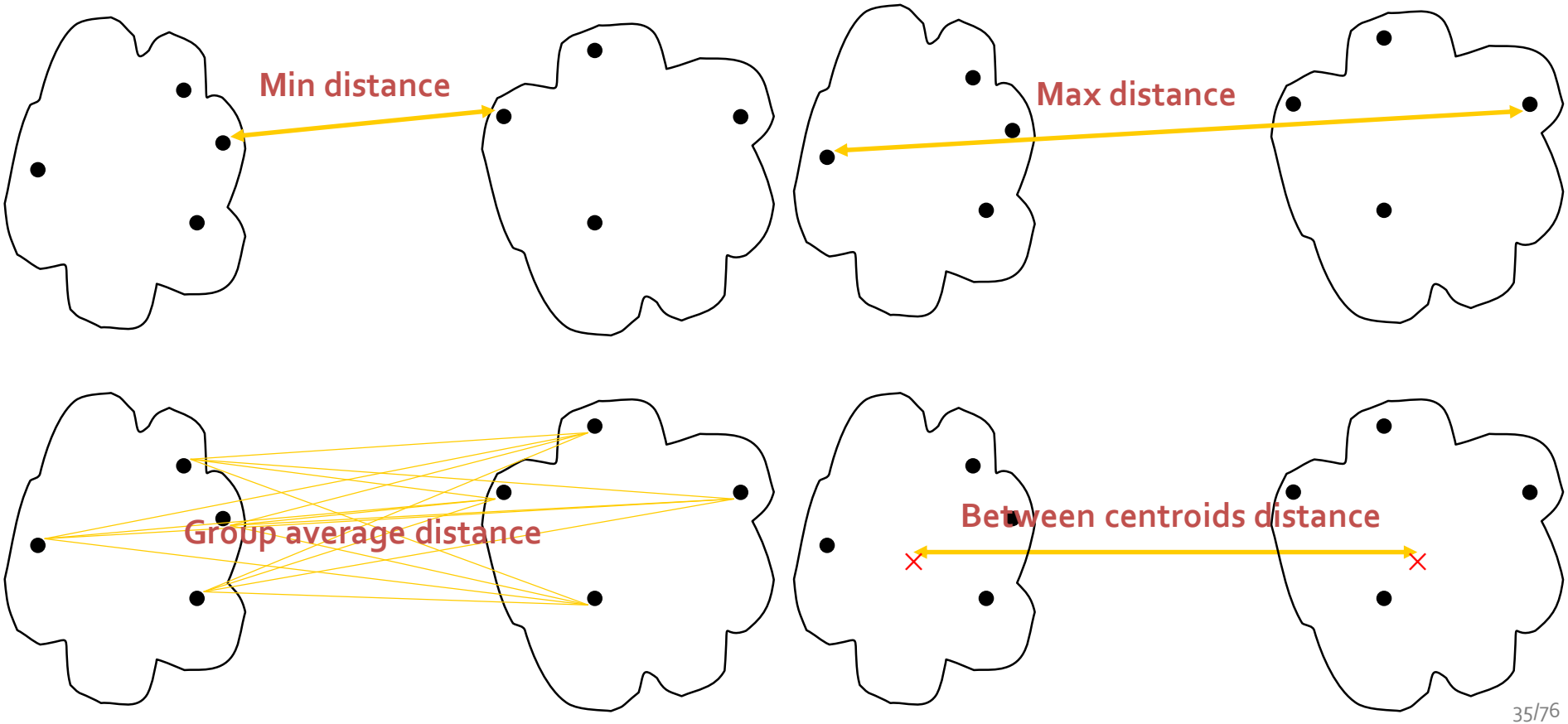
❖ 계층적 군집화의 두 가지 방식

- 상향식 군집화: Agglomerative clustering
  - ✓초기에 모든 개체들을 개별적인 군집으로 가정
  - ✓각 단계에서 유사한 개체/군집 결합 → 모든 개체들이 하나의 군집으로 통합되면 완료
- 하향식 군집화: Divisive clustering
  - ✓모든 개체가 하나의 군집으로 이루어진 상태에서 출발
  - ✓각 단계에서 가장 유의미하게 구분되는 지점을 판별하여 지속적으로 데이터를 분할

계층적 군집화: Hierarchical Clustering

❖ 상향식 군집화 알고리즘

- 핵심 수행 절차: 두 군집 사이의 유사도/거리 측정
  - ✓ Min, max, group average, between centroid, etc.

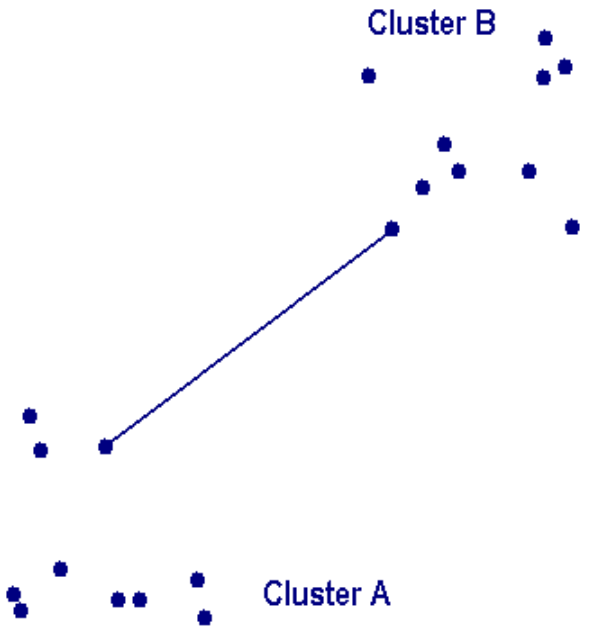


계층적 군집화: Hierarchical Clustering

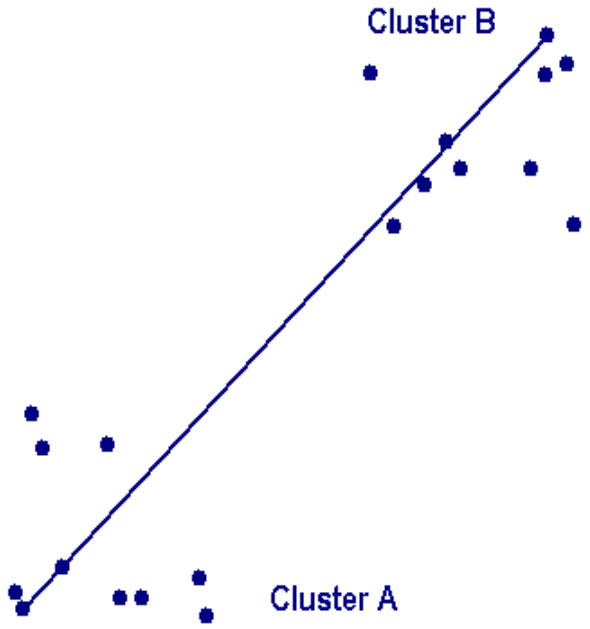
❖ 상향식 군집화 알고리즘

- Single linkage (minimum distance): 각 군집에 속한 개체들 사이의 거리 중 가장 가까운 값을 군집간 거리로 정의
- Complete linkage (maximum distance): 각 군집에 속한 개체들 사이의 거리 중 가장 먼 값을 군집간 거리로 정의

Single linkage



Complete linkage

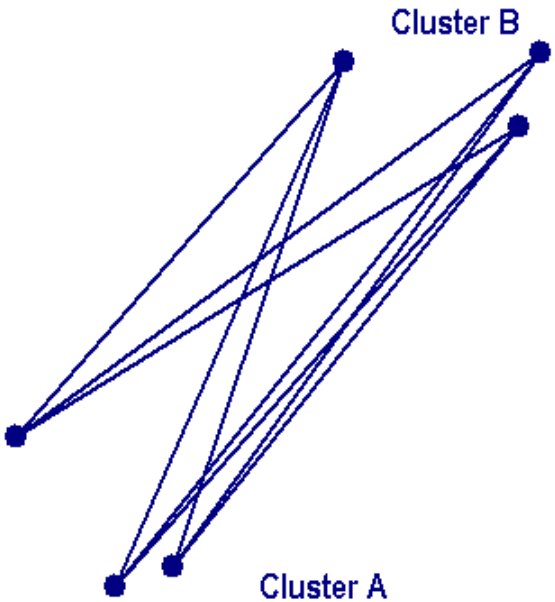


계층적 군집화: Hierarchical Clustering

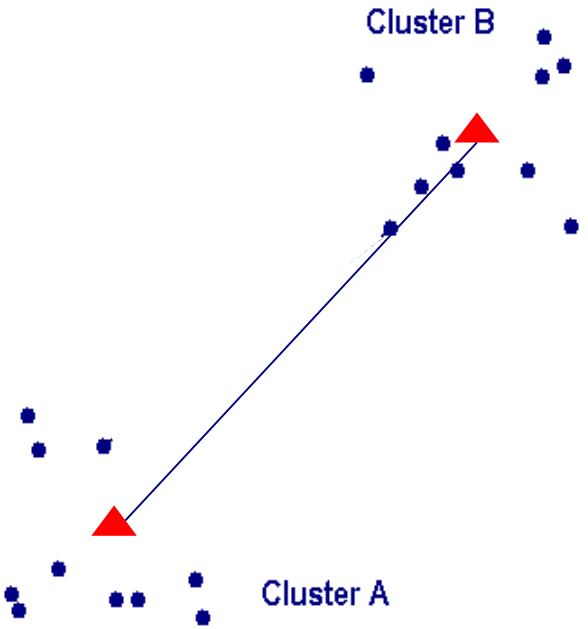
❖ 상향식 군집화 알고리즘

- Average linkage (mean distance): 각 군집에 속한 개체들 사이의 거리 평균값을 군집간 거리로 정의
- Centroid linkage (distance between centroids): 각 군집의 중심간 거리를 군집간 거리로 정의

Average linkage



Centroid linkage



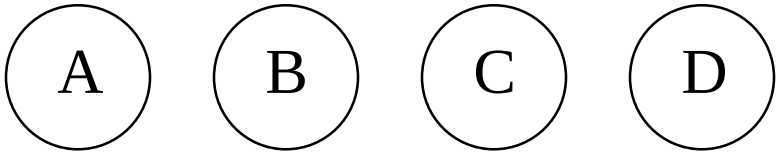
계층적 군집화: Hierarchical Clustering

- 상향식 군집화 알고리즘
  - 1단계: 모든 개체를 개별 군집으로 정의하고 군집간 거리 행렬 계산
  - 2단계: 다음 절차를 반복
    - ✓2-1단계: 가장 가까운 두 개의 군집을 하나의 군집으로 통합
    - ✓2-2단계: 군집간 거리 행렬 업데이트
  - 종료 조건: 모든 개체가 하나의 군집으로 통합되면 종료

계층적 군집화: Hierarchical Clustering

❖ 계층적 군집화 절차 예시

Initial Data Items



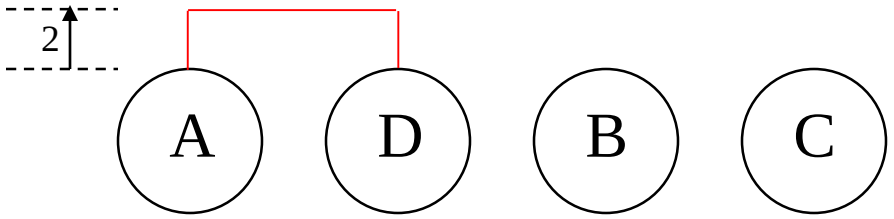
Distance Matrix

Dist	A	B	C	D
A		20	7	2
B			10	25
C				3
D				

계층적 군집화: Hierarchical Clustering

❖ 계층적 군집화 절차 예시

Current Clusters



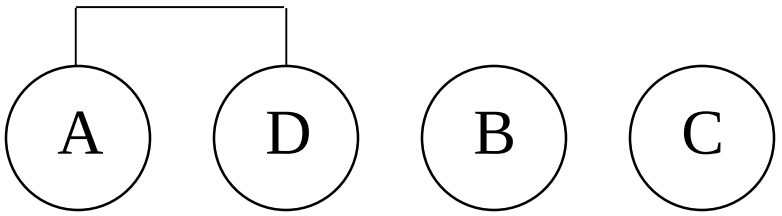
Distance Matrix

Dist	A	B	C	D
A		20	7	2
B			10	25
C				3
D				

계층적 군집화: Hierarchical Clustering

❖ 계층적 군집화 절차 예시

Current Clusters



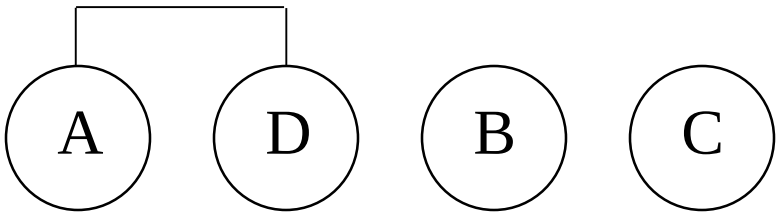
Distance Matrix

Dist	AD	B	C	
AD		20	3	
B			10	
C				

계층적 군집화: Hierarchical Clustering

❖ 계층적 군집화 절차 예시

Current Clusters



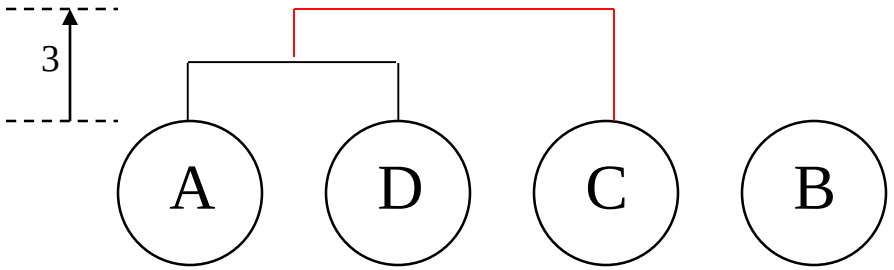
Distance Matrix

Dist	AD	B	C	
AD		20	3	
B			10	
C				

계층적 군집화: Hierarchical Clustering

❖ 계층적 군집화 절차 예시

Current Clusters



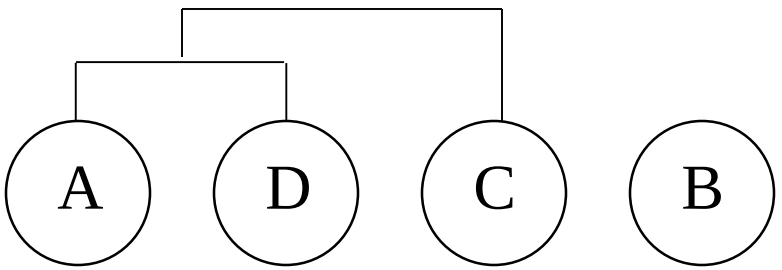
Distance Matrix

Dist	AD	B	C	
AD		20	3	
B			10	
C				

계층적 군집화: Hierarchical Clustering

❖ 계층적 군집화 절차 예시

Current Clusters



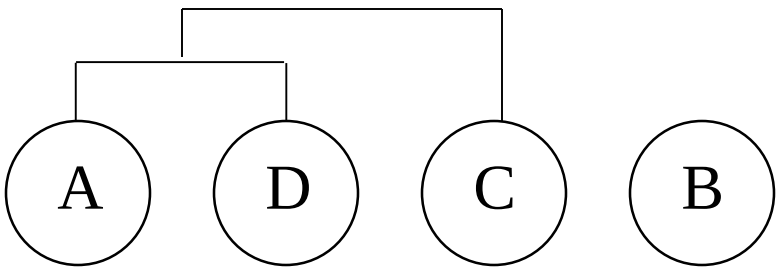
Distance Matrix

Dist	AD C	B		
AD C		10		
B				

계층적 군집화: Hierarchical Clustering

❖ 계층적 군집화 절차 예시

Current Clusters



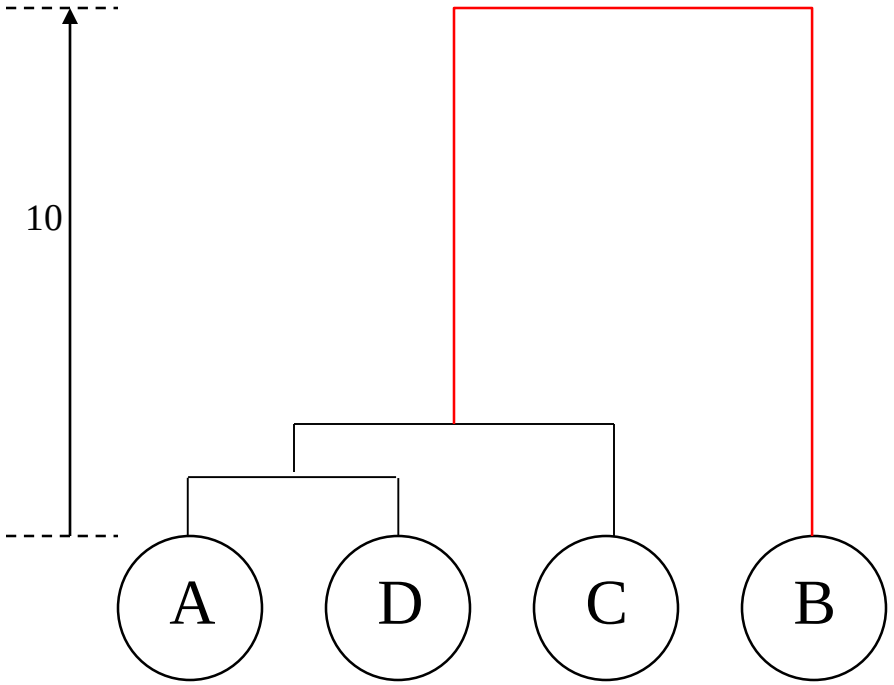
Distance Matrix

Dist	AD C	B		
AD C		10		
B				

계층적 군집화: Hierarchical Clustering

❖ 계층적 군집화 절차 예시

Current Clusters



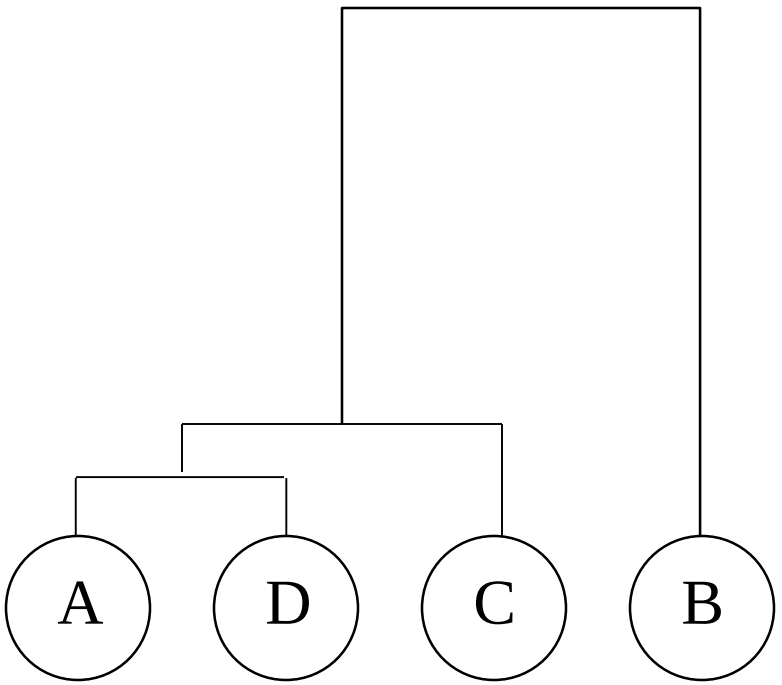
Distance Matrix

Dist	AD C	B		
AD C		10		
B				

계층적 군집화: Hierarchical Clustering

❖ 계층적 군집화 절차 예시

Final Result

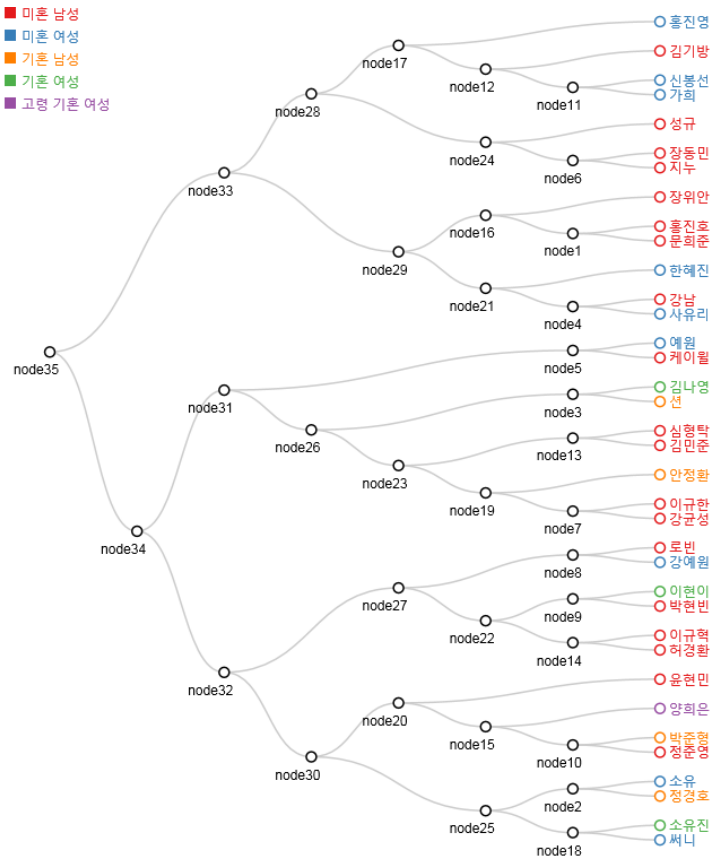


Distance Matrix

Dist	AD CB			
AD CB				

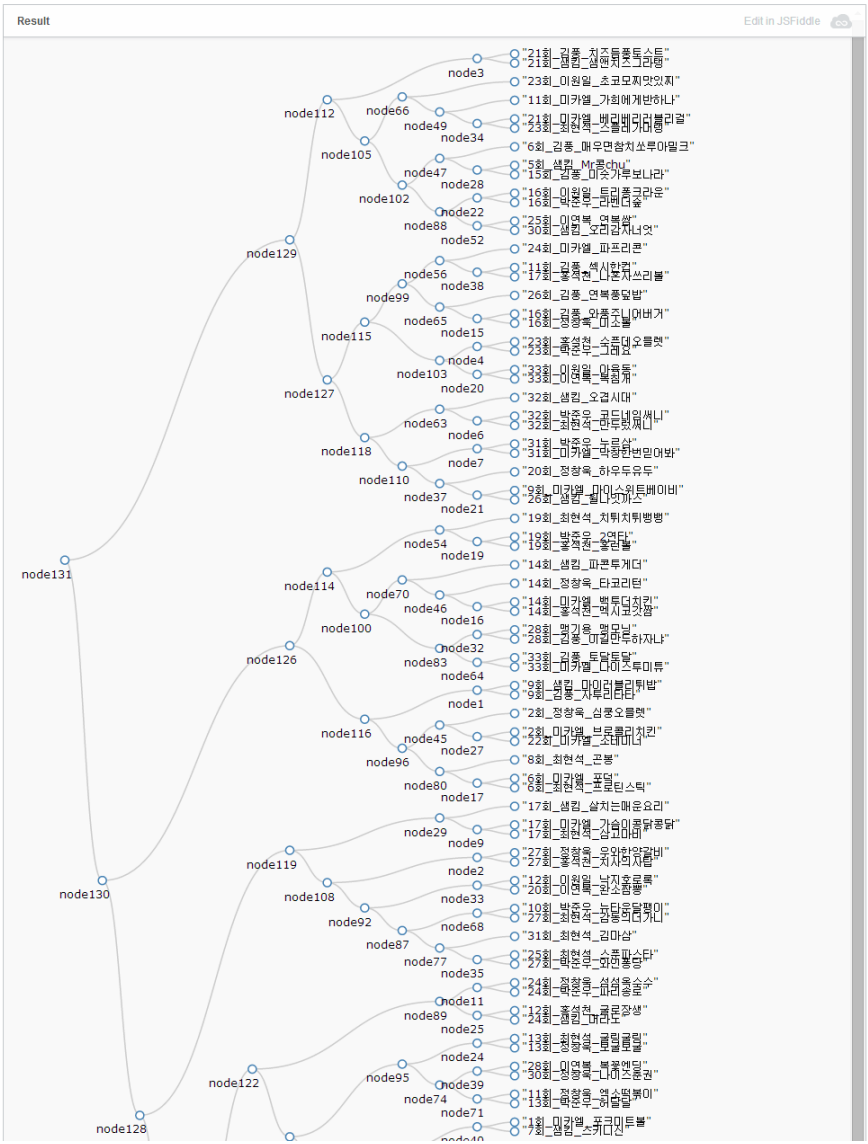
계층적 군집화: Hierarchical Clustering

❖ 냉장고를 부탁해



냉장고 재료를 이용한 게스트 군집화

레시피 계층적 군집화 분석

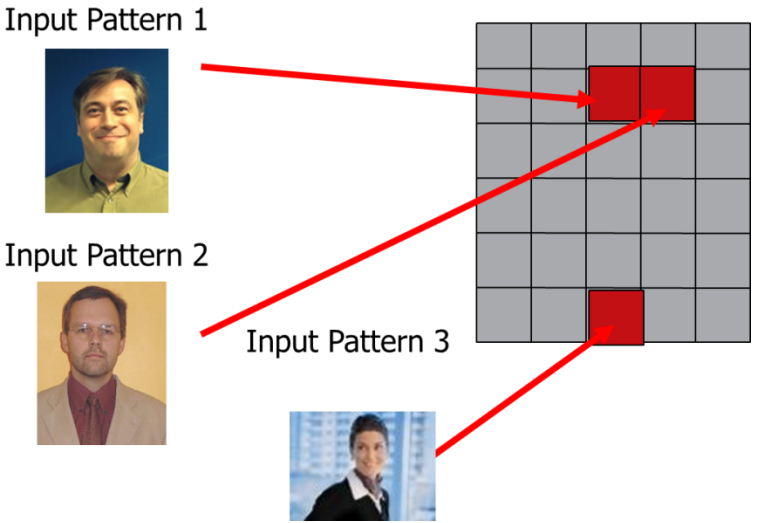


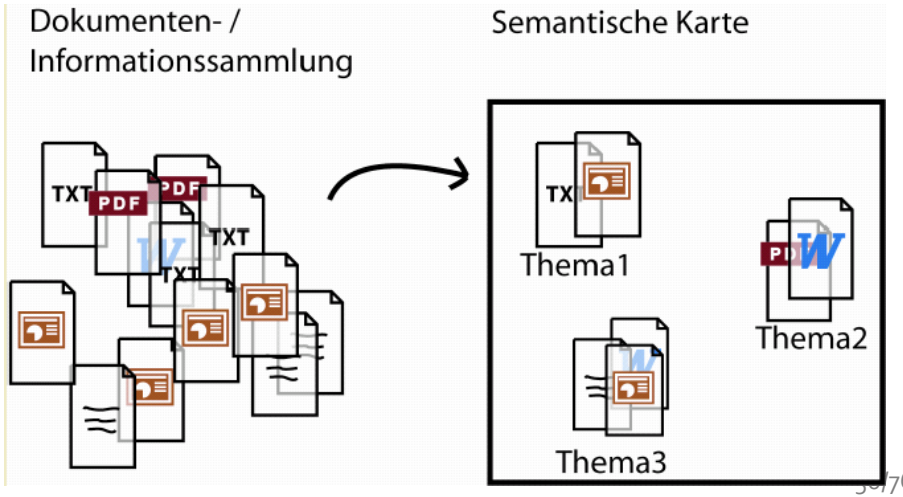
목차

자기조직화 지도: Self-Organizing Map (SOM)

❖ 자기 조직화 지도: Self-Organizing Map (SOM)

- 고차원의 데이터를 사람이 시각적으로 이해할 수 있는 저차원(2차원 또는 3차원) 격자에 표현하는 방식
  - ✓고차원에서 유사한 개체들은 저차원에 인접한 격자들과 연결됨
  - ✓인공신경망 학습 알고리즘을 차용: 비지도적 경쟁학습
- 저차원의 격자에서의 유사도는 고차원 입력 공간에서의 유사도를 최대한 보존하도록 학습

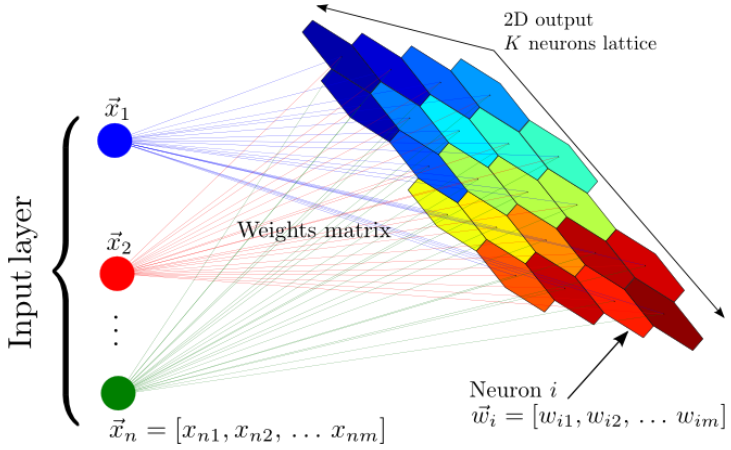


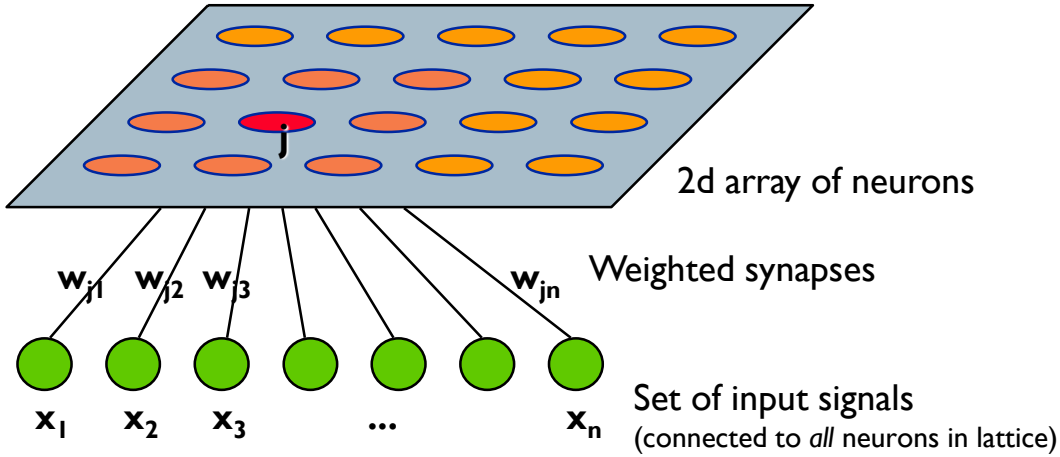


자기조직화 지도: Self-Organizing Map (SOM)

❖ 자기조직화 지도: 구조

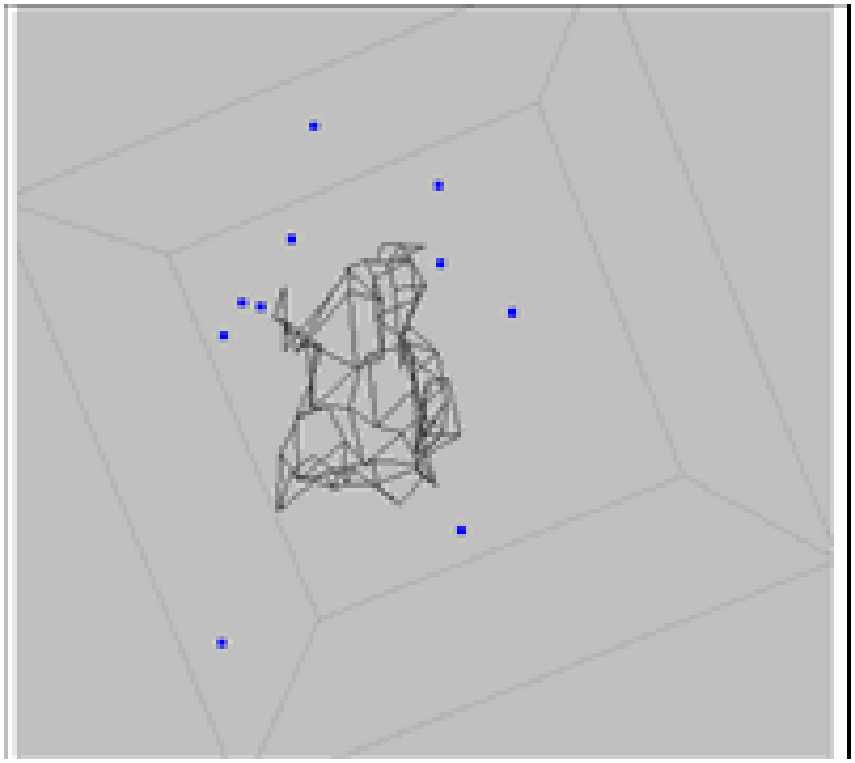
- 저차원 격자의 모든 노드는 원 공간의 모든 개체들과 가중치 w로 연결되어 있음
- 저차원 격자의 노드들은 서로 위치적인 유사도 관계를 가짐
- 원 공간의 각 개체와 가장 유사한 형태의 가중치를 갖는 Winning 노드가 선택됨
- 선택된 노드 및 근처 노드들이 활성화되어 원 공간의 개체와 유사하도록 가중치를 조정함

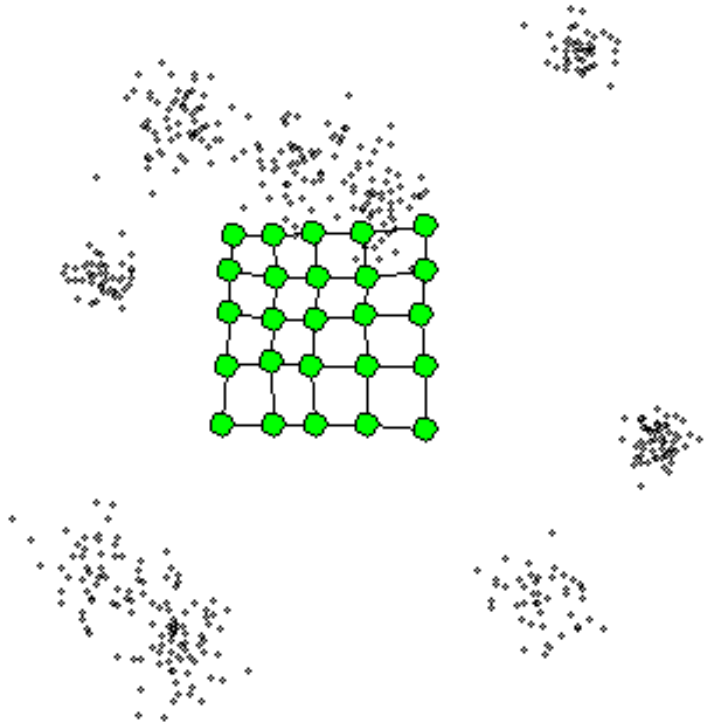




자기조직화 지도: Self-Organizing Map (SOM)

❖ SOM 예시





자기조직화 지도: Self-Organizing Map (SOM)

Components of Self-Organization

1. Initialization
2. Competition
3. Cooperation
4. Adaptation

자기조직화 지도: Self-Organizing Map (SOM)

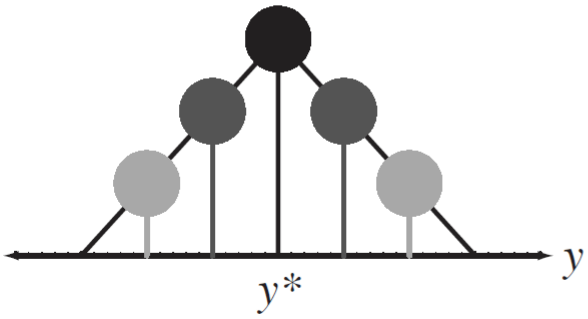
❖ 자기조직화지도: 학습

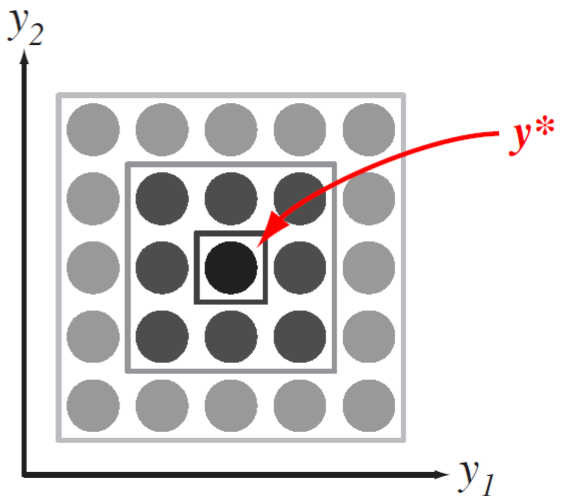
1. 자기조직화지도 격자 설정
2. 각 노드의 가중치를 설정 (usually at random)
3. 학습 데이터의 한 개체에 대해 모든 격자와의 유사도를 평가하여 Best Machine Unit (BMU) 선택
4. BMU와 이웃 노드와의 유사성 계산
5. BMU는 학습 데이터와 유사하도록 가중치를 업데이트하고, 이웃노드들도 일정 수준 가중치 업데이트를 수행
6. 가중치의 변화가 없을 때까지 Step 3부터 Step 5까지를 반복

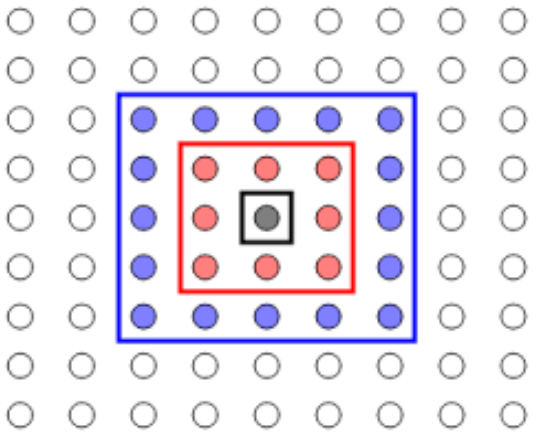
자기조직화 지도: Self-Organizing Map (SOM)

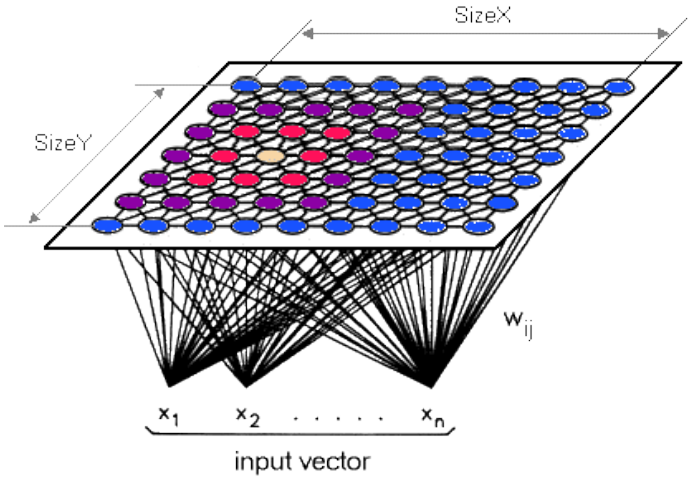
❖ 자기조직화지도: 학습

4. 이웃노드와의 유사성 계산









자기조직화 지도: Self-Organizing Map (SOM)

❖ 자기조직화지도: 학습

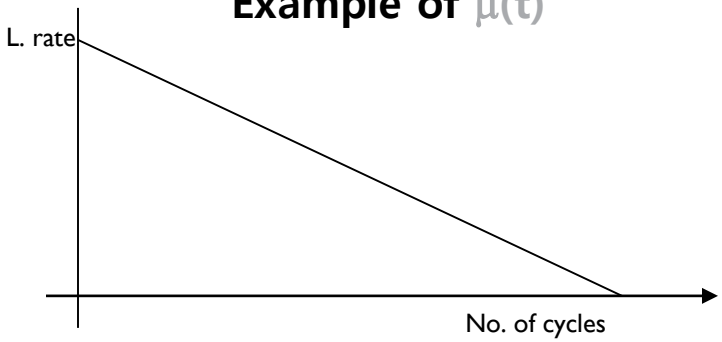
5. BMU는 학습 데이터와 유사하도록 가중치를 업데이트하고, 이웃노드들도 일정 수준 가중치 업데이트를 수행

<SOM Weight Update Equation>

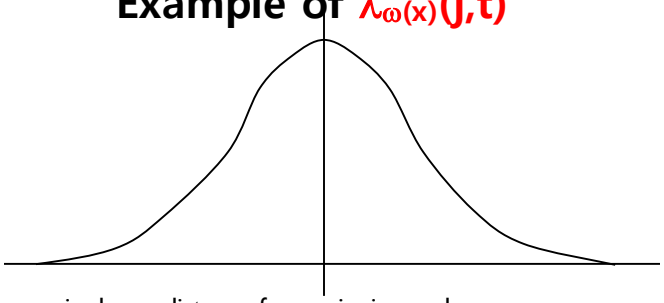
$$w_j(t+1) = w_j(t) + \mu(t)\,\lambda_{\omega(x)}(j,t)\,[x - w_j(t)]$$

"The weights of every node are updated at each cycle by adding
Current learning rate × Degree of neighbourhood with respect to winner × Difference between current weights and input vector
to the current weights"

Example of $\mu(t)$



Example of $\lambda_{\omega(x)}(j,t)$



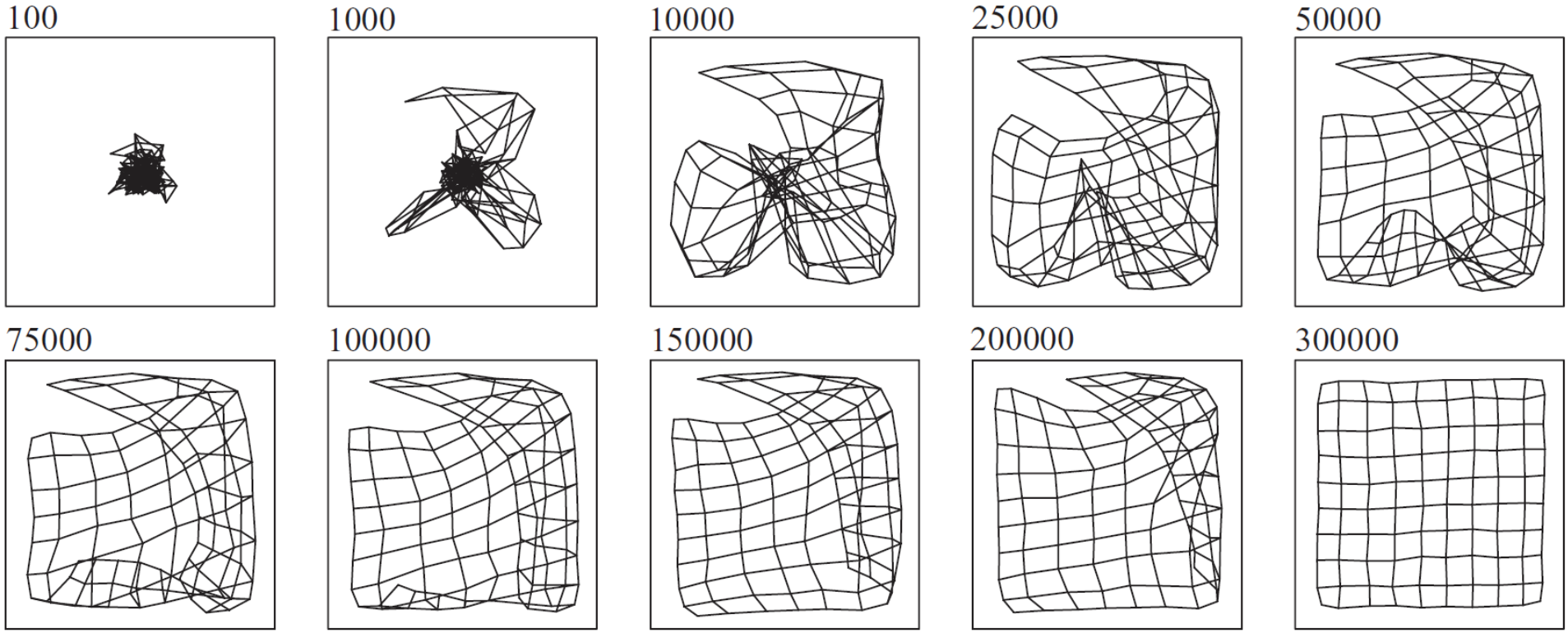
–x-axis shows distance from winning node

–y-axis shows 'degree of neighbourhood' (max. 1)

자기조직화 지도: Self-Organizing Map (SOM)

❖ 자기조직화 지도: 수렴

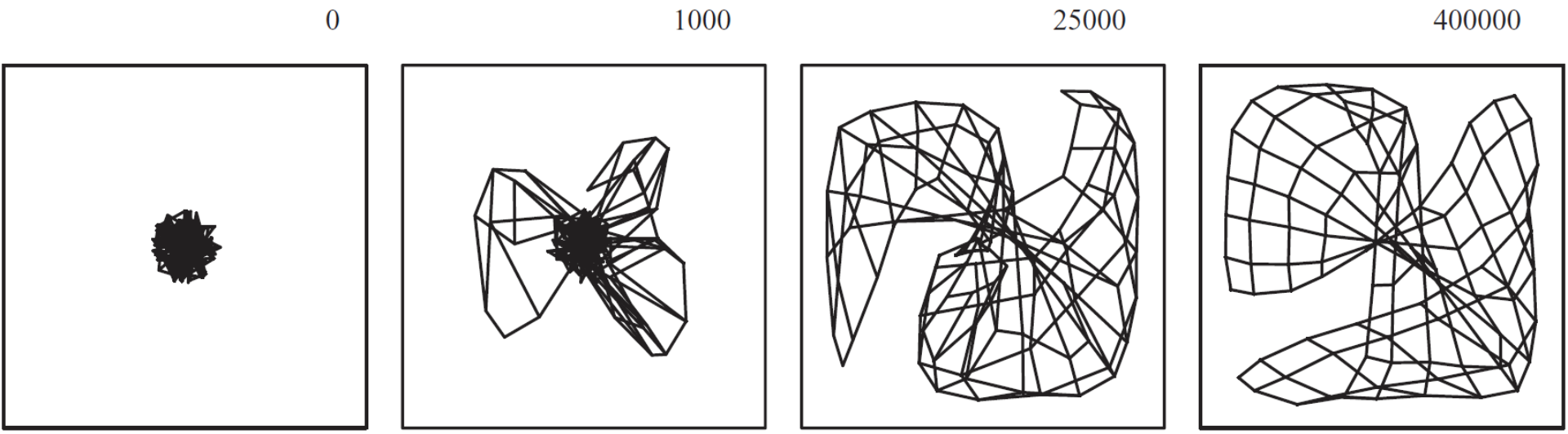
- 행복하게도...



자기조직화 지도: Self-Organizing Map (SOM)

❖ 자기조직화 지도: 수렴

- 가끔씩은...



- 초기 가중치를 재설정

목차

K-평균 군집화

- K-평균 군집화는 다양한 패키지에서 제공
  - stats, kml, kml3d, RSKC, skmeans, sparcl, etc.
- R에서 기본 제공되는 Iris 데이터를 이용한 K-평균 군집화 수행

```
# Package for cluster validity
install.packages("clValid")
install.packages("plotrix")

library(clValid)
library(plotrix)

# Load the Iris dataset
data(iris)

# Part 1: K-Means Clustering ---------------------------------------------
# Remove the class label
newiris <- iris
newiris$Species <- NULL
rownames(newiris) <- paste("I", 1:150, sep = "_")

# Perform K-Means Clustering with K=3
kc <- kmeans(newiris,3)

str(kc)
kc$centers
kc$size
kc$cluster
```

K-평균 군집화

❖ 결과 확인

- 각 군집의 중심, 군집에 속한 개체 수, 각 개체가 속한 군집 id 등

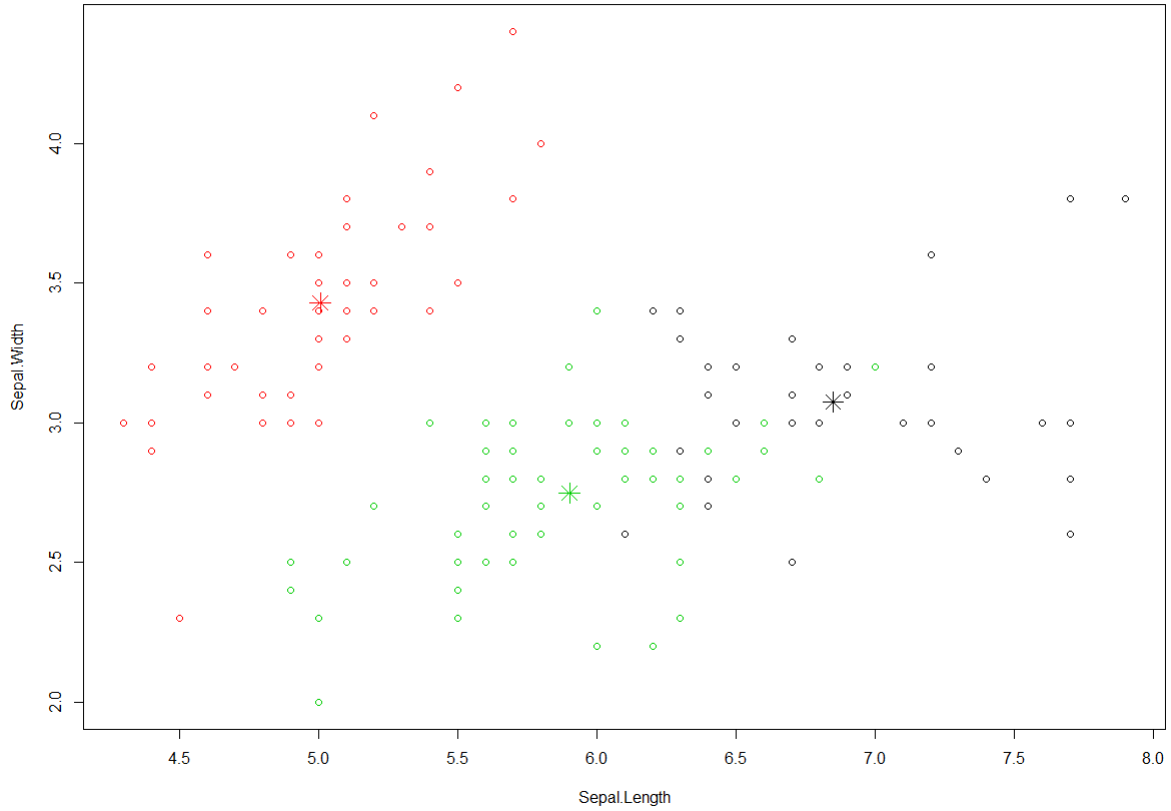
```
> str(kc)
List of 9
 $ cluster     : int [1:150] 2 2 2 2 2 2 2 2 2 2 ...
 $ centers     : num [1:3, 1:4] 6.85 5.01 5.9 3.07 3.43 ...
  ..- attr(*, "dimnames")=List of 2
  .. ..$ : chr [1:3] "1" "2" "3"
  .. ..$ : chr [1:4] "Sepal.Length" "Sepal.Width" "Petal.Length" "Petal.Width"
 $ totss       : num 681
 $ withinss    : num [1:3] 23.9 15.2 39.8
 $ tot.withinss: num 78.9
 $ betweenss   : num 603
 $ size        : int [1:3] 38 50 62
 $ iter        : int 2
 $ ifault      : int 0
 - attr(*, "class")= chr "kmeans"
> kc$centers
  Sepal.Length Sepal.Width Petal.Length Petal.Width
1     6.850000    3.073684     5.742105    2.071053
2     5.006000    3.428000     1.462000    0.246000
3     5.901613    2.748387     4.393548    1.433871
> kc$size
[1] 38 50 62
> kc$cluster
  [1] 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 3 3 1 3 3 3 3 3 3 3 3
 [62] 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 1 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 1 3 1 1 1 1 3 1 1 1 1 1 3 3 1 1 1 1 3 1 3
[123] 1 3 1 1 3 3 1 1 1 1 1 3 1 1 1 1 3 1 1 1 3 1 1 1 3
> # Compare the assigned clusters and the Species
> table(iris$Species, kc$cluster)

              1  2  3
  setosa      0 50  0
  versicolor  2  0 48
  virginica  36  0 14
```

K-평균 군집화

❖ 군집화 결과 도시

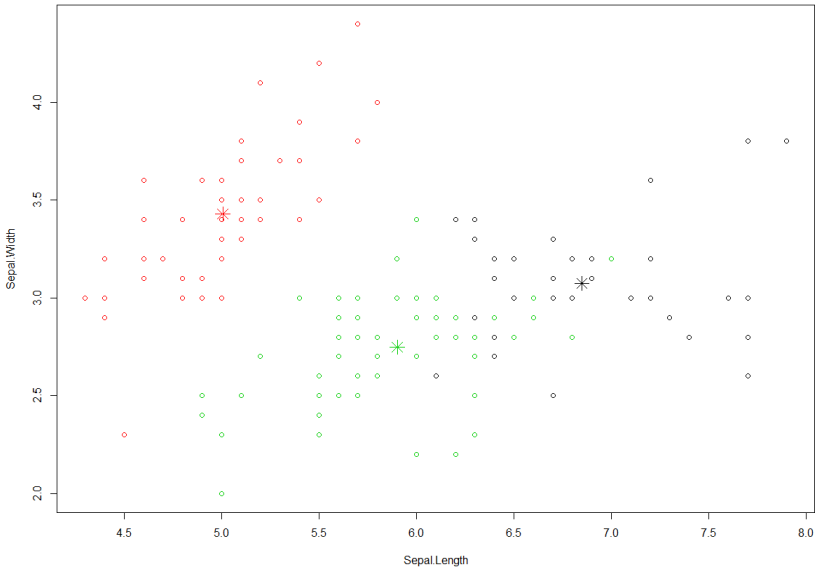
```
28 plot(newiris[,c("Sepal.Length", "Sepal.Width")], col = kc$cluster)
29 points(kc$centers[,c("Sepal.Length", "Sepal.Width")], col = 1:3, pch = 8, cex=2)
```

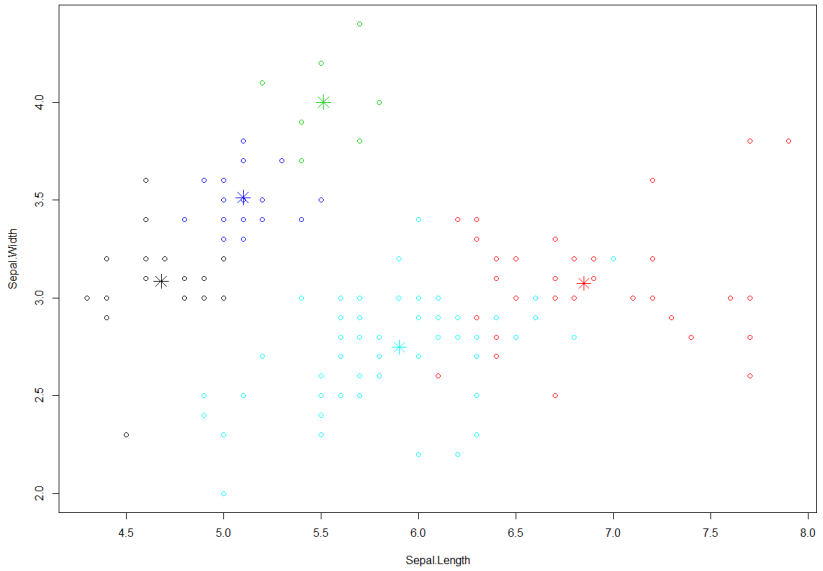


K-평균 군집화

❖ 군집의 수를 5로 다시 설정하고 K-평균 군집화 수행

```
31  # Perform K-Means Clustering with K=5
32  kc <- kmeans(newiris,5)
33
34  # Compare the assigned clusters and the Species
35  table(iris$Species, kc$cluster)
36
37  plot(newiris[,c("Sepal.Length", "Sepal.Width")], col = kc$cluster)
38  points(kc$centers[,c("Sepal.Length", "Sepal.Width")], col = 1:5, pch = 8, cex=2)
```





K-평균 군집화

❖ 군집 타당성 지표 비교

```
40 # Evaluating the cluster validity measures
41 newiris.clValid <- clValid(newiris, 2:10, clMethods = "kmeans",
42                           validation = c("internal", "stability"))
43 summary(newiris.clValid)
```

```
> summary(newiris.clValid)

Clustering Methods:
 kmeans

Cluster sizes:
 2 3 4 5 6 7 8 9 10

Validation Measures:
                         2       3       4       5       6       7       8       9      10

kmeans APN           0.0130  0.0630  0.1572  0.2394  0.1680  0.1954  0.2212  0.2198  0.2619
       AD            1.2223  0.9390  0.8722  0.8149  0.7309  0.6946  0.6804  0.6489  0.6306
       ADM           0.0562  0.1131  0.2803  0.3316  0.2293  0.2340  0.2523  0.2245  0.2593
       FOM           0.4990  0.3935  0.3590  0.3534  0.3354  0.3144  0.3131  0.3050  0.3009
       Connectivity  6.1536 10.0917 17.5194 27.9373 36.4873 33.9595 38.9556 49.9901 58.0988
       Dunn          0.0765  0.0988  0.1365  0.0823  0.0853  0.0872  0.0872  0.0617  0.0684
       Silhouette    0.6810  0.5528  0.4981  0.4887  0.3648  0.3609  0.3556  0.3360  0.3391

Optimal Scores:

             Score  Method Clusters
APN          0.0130 kmeans 2
AD           0.6306 kmeans 10
ADM          0.0562 kmeans 2
FOM          0.3009 kmeans 10
Connectivity 6.1536 kmeans 2
Dunn         0.1365 kmeans 4
Silhouette   0.6810 kmeans 2
```

계층적 군집화

❖ Personal Loan 데이터를 이용하여 은행 고객 군집화

Data Description:

ID	Customer ID
Age	Customer's Age in completed years
Experience	#years of professional experience
Income	Annual income of the customer ($000)
ZIPCode	Home Address ZIP code.
Family	Family size (dependents) of the customer
CCAvg	Avg. Spending on Credit Cards per month ($000)
Education	Education Level. 1: Undergrad; 2: Graduate; 3: Advanced/Professional
Mortgage	Value of house mortgage if any. ($000)
Personal Loan	Did this customer accept the personal loan offered in the last campaign?
Securities Account	Does the customer have a Securities account with the bank?
CD Account	Does the customer have a Certificate of Deposit (CD) account with the bank?
Online	Does the customer use internet banking facilities?
CreditCard	Does the customer use a credit card issued by UniversalBank?

계층적 군집화

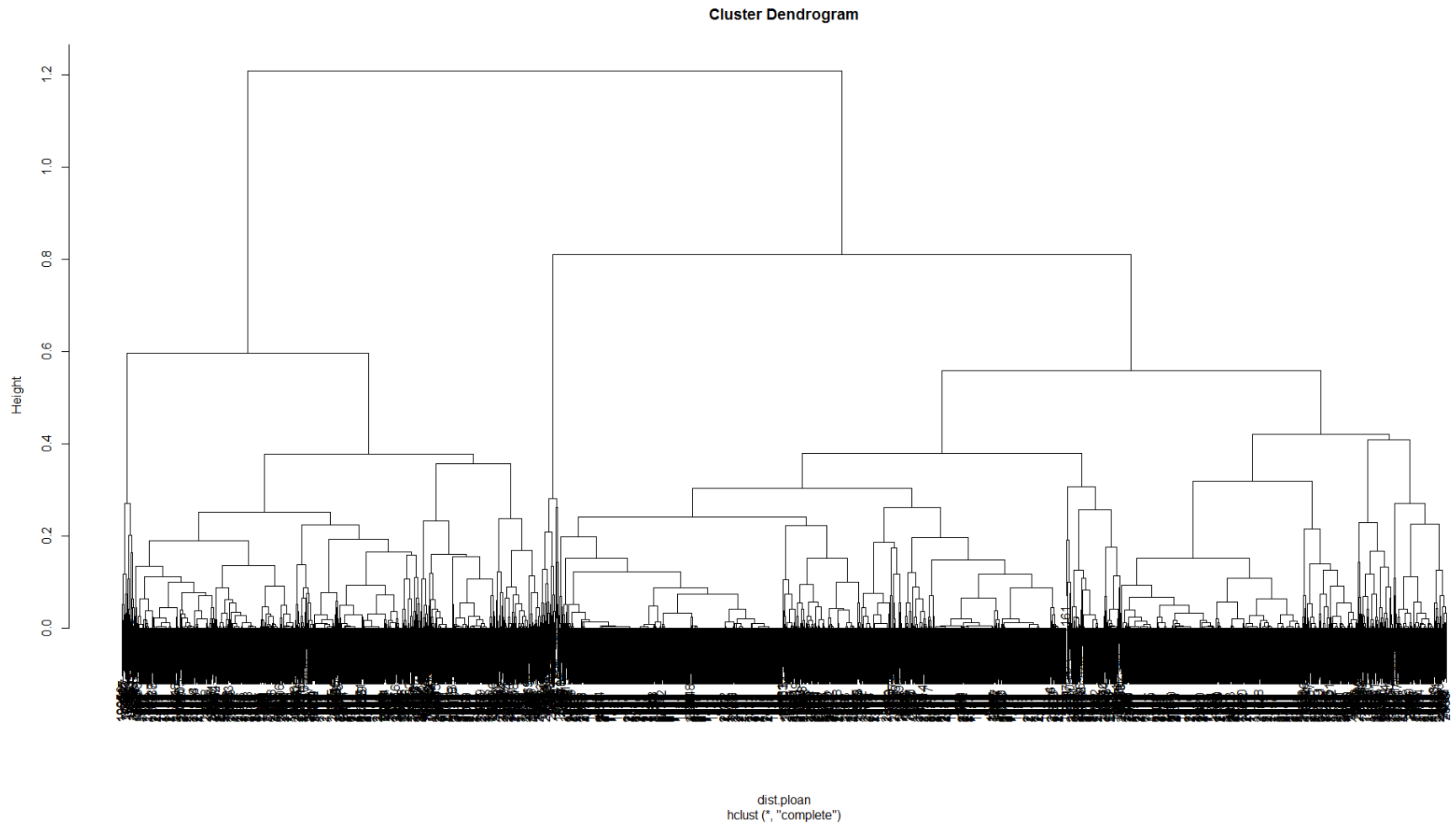
❖ Personal Loan 데이터를 이용하여 은행 고객 군집화

- 고객간의 유사도 측정은 피어슨 상관계수 사용
- 계층적 군집화를 위한 집단간 거리는 complete linkage 사용

```
45 # Part 2: Hierarchical Clustering ----------------------------------------
46 ploan <- read.csv("Personal Loan.csv")
47 ploan.x <- ploan[,-c(1,5,10)]
48 ploan.x.scaled <- scale(ploan.x, center = TRUE, scale = TRUE)
49
50 # Compute the similarity using the spearman coefficient
51 cor.Mat <- cor(t(ploan.x.scaled), method = "spearman")
52 dist.ploan <- as.dist(1-cor.Mat)
53
54 # Perform hierarchical clustering
55 hr <- hclust(dist.ploan, method = "complete", members=NULL)
```

계층적 군집화

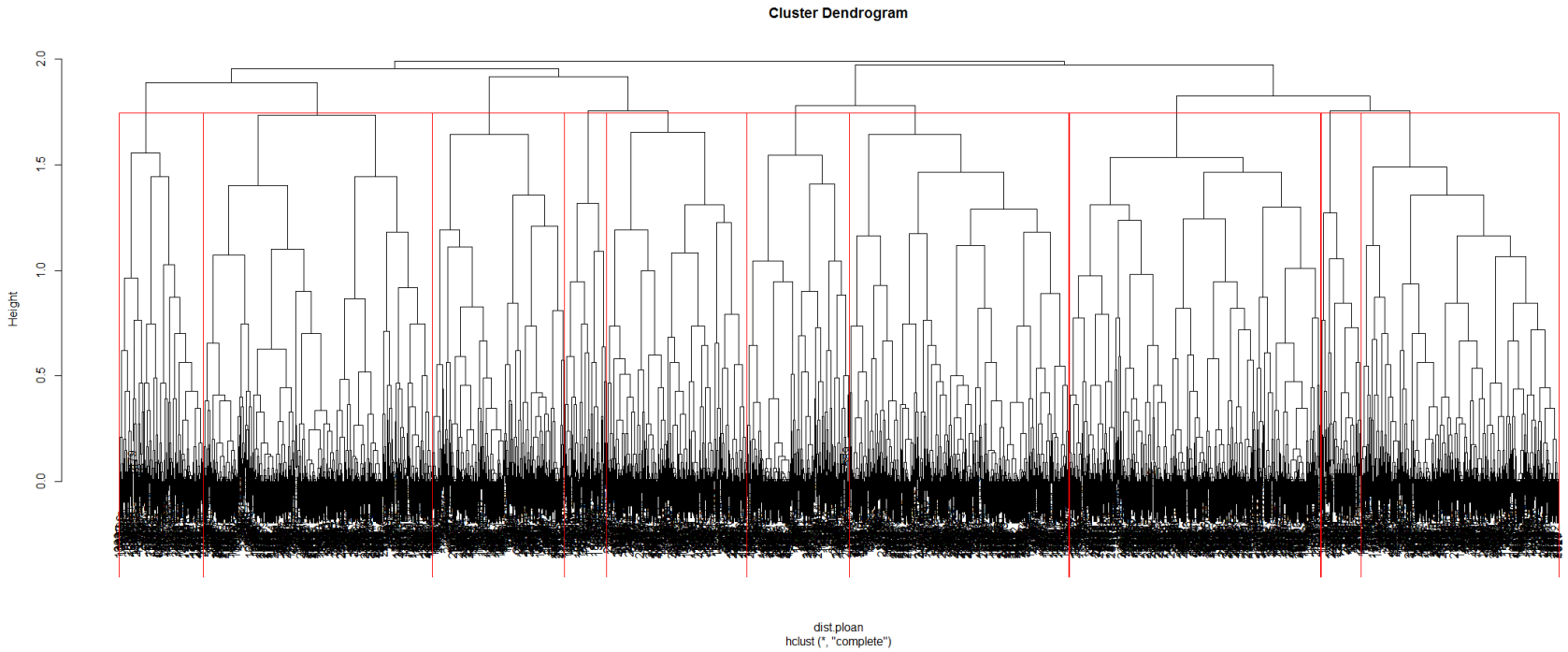
❖ 덴드로그램 생성



계층적 군집화

❖ 군집화 수행

```
57 # plot the results
58 plot(hr)
59 plot(hr, hang = -1)
60 plot(as.dendrogram(hr), edgePar=list(col=3, lwd=4), horiz=T)
61
62 # Find the clusters
63 mycl <- cutree(hr, k=10)
64 mycl
```



계층적 군집화

❖ 각 군집의 특성 분석

```
70 # Compare each cluster for HC
71 cluster.hc <- data.frame(ploan.x.scaled, ploanYN = ploan[,10],
72                           clusterID = as.factor(mycl))
73 hc.summary <- data.frame()
74
75 for (i in 1:(ncol(cluster.hc)-1)){
76   hc.summary = rbind(hc.summary,
77                      tapply(cluster.hc[,i], cluster.hc$clusterID, mean))
78 }
79
80 colnames(hc.summary) <- paste("cluster", c(1:10))
81 rownames(hc.summary) <- c(colnames(ploan.x), "LoanRatio")
82 hc.summary
```

계층적 군집화

❖ 각 군집의 특성 분석

	cluster 1	cluster 2	cluster 3	cluster 4	cluster 5	cluster 6	cluster 7	cluster 8	cluster 9	cluster 10
Age	-0.978375630	0.62768747	-1.00065813	0.45556464	0.623411101	0.3606053	0.83631077	-0.94640912	0.23039152	-0.27857180
Experience	-0.984955242	0.63865510	-1.01149000	0.45826338	0.594577161	0.3583185	0.84354883	-0.93187619	0.22877326	-0.27319032
Income	-0.041720588	0.11353167	-0.34249916	0.67417711	0.055576655	0.2654431	-0.76794149	0.68688171	-0.30617789	0.99685044
Family	0.249749572	-0.06963678	0.64165613	-0.81485549	0.758429117	-0.2598958	0.09881633	-0.61816862	-0.84443724	-0.42263030
CCAvg	0.056049820	0.07592153	-0.27792518	0.55790241	0.002870789	0.3816039	-0.55877036	0.47138980	-0.43458985	0.01431525
Education	0.219734047	-0.33718946	0.21399718	-0.13166151	0.168847347	0.0920721	0.02249967	-0.20870083	0.82666013	-0.77272322
Mortgage	-0.109905334	-0.04520828	-0.09913763	0.04240734	0.412406634	-0.3300013	-0.16572200	0.12002376	0.89665284	1.20219168
Securities.Account	0.002768788	0.02273880	0.04403728	0.13838377	-0.175403903	-0.3507727	0.13171186	0.13469351	-0.12216694	-0.21924611
CD.Account	-0.197847765	-0.21063348	-0.07349502	-0.04132722	-0.229615630	0.2672237	0.03286815	0.48516412	-0.08077930	0.98776848
Online	-1.200618309	-1.20906773	0.76690876	0.81151795	0.624917820	0.3859185	0.51306132	0.67151852	-1.10189473	0.42941455
CreditCard	-0.044161882	-0.17087863	-0.13667044	-0.59150572	-0.639594361	1.5628656	-0.25222704	-0.05804595	0.11553475	1.26115872
LoanRatio	0.098837209	0.10755149	0.07874016	0.17467249	0.102739726	0.1147541	0.01005025	0.21348315	0.07142857	0.20547945

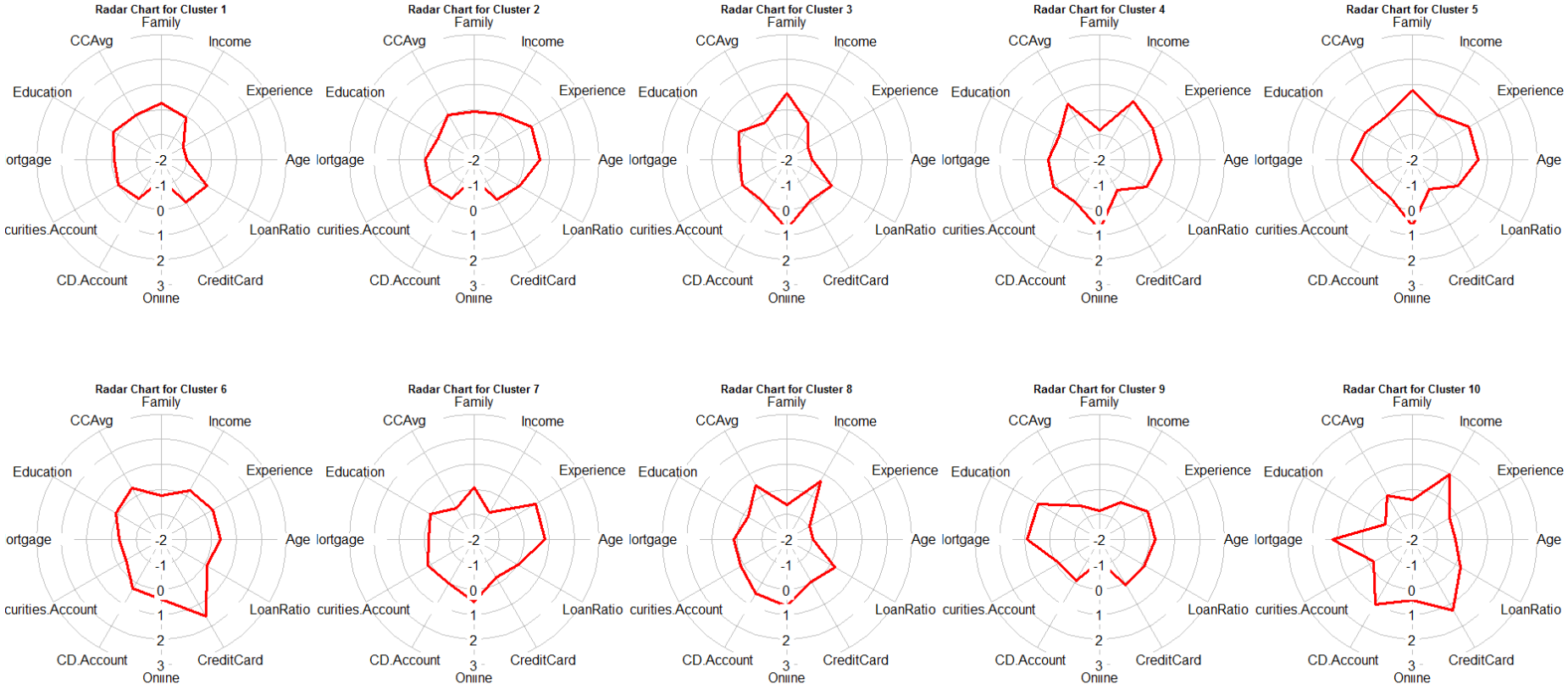
계층적 군집화

❖ 각 군집의 특성 분석

```
84  # Radar chart
85  par(mfrow = c(2,5))
86  for (i in 1:10){
87    plot.title <- paste("Radar Chart for Cluster", i, sep=" ")
88    radial.plot(hc.summary[,i], labels = rownames(hc.summary),
89                radial.lim=c(-2,3), rp.type = "p", main = plot.title,
90                line.col = "red", lwd = 3, show.grid.labels=1)
91  }
92  dev.off()
```

계층적 군집화

❖ 각 군집의 특성 분석



자기조직화지도: SOM

- SOM을 수행할 수 있는 패키지
  - kohonen, som, wccsom, etc.
- som/kohonen packages 사용
- 대상 데이터: Yeast & Wine
  - Yeast: 6601 genes measured at 18 time points
  - Wines: 177 records with 13 characteristics

자기조직화지도: SOM

❖ Package som

```
# Part 3: Self-Organizing Map ----------------------------------------
# som package install
install.packages("som", dependencies = TRUE)
detach("package:kohonen", unload=TRUE)
library(som)

# Load the yeast dataset
data(yeast)
yeast <- yeast[, -c(1, 11)]
yeast <- normalize(yeast, byrow=FALSE)

# Train SOM with two different settings
som1 <- som(yeast, xdim=5, ydim=5, topol="rect", neigh="gaussian")
som2 <- som(yeast, xdim=5, ydim=5, topol="hexa", neigh="bubble")

# See the results
summary(som1)
plot(som1)
som1$visual[1:10,]

summary(som2)
plot(som2)
som2$visual[1:10,]
```

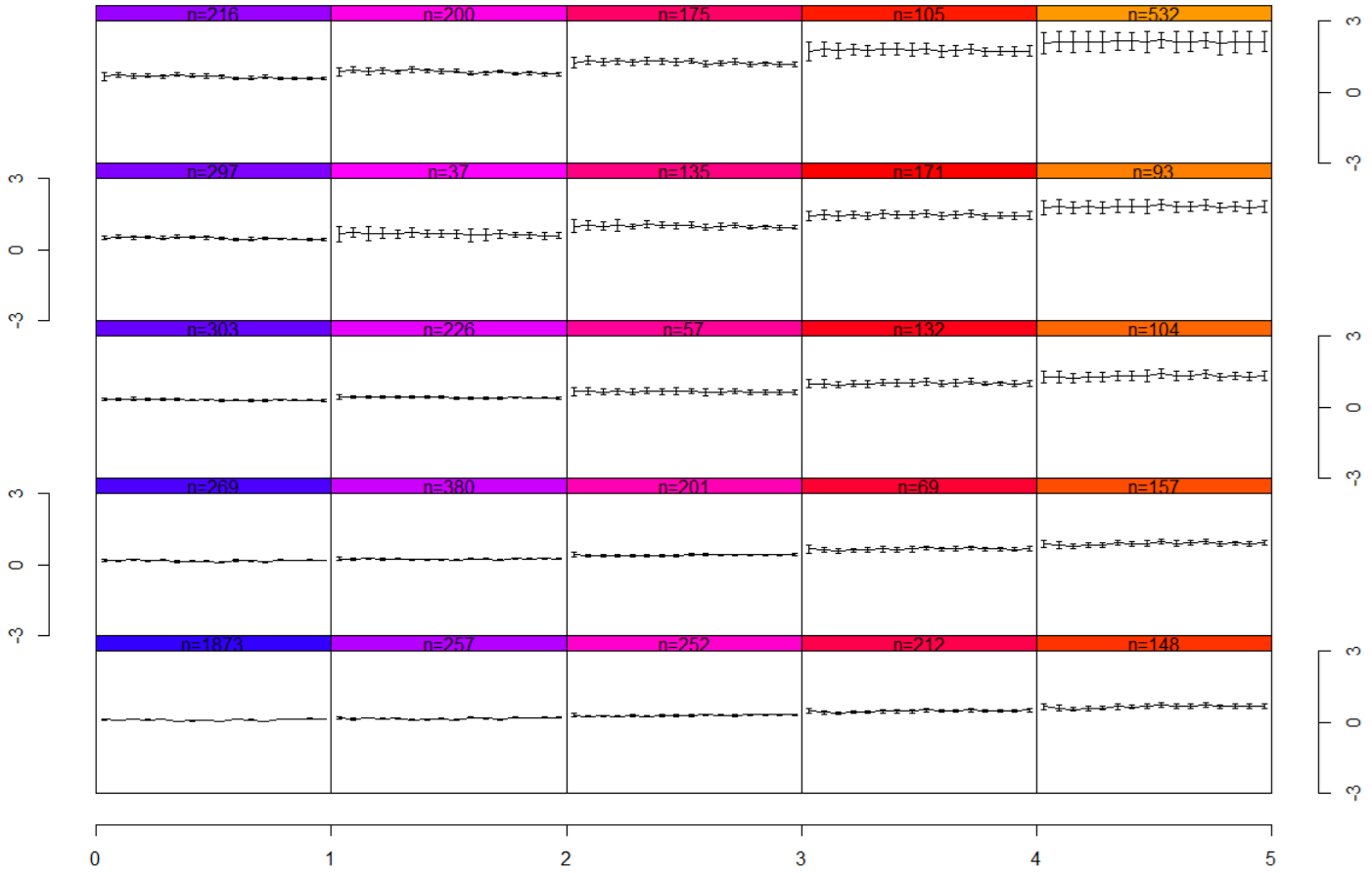
자기조직화지도: SOM

❖ Package som

```
> summary(som1)
Initialization:  linear
Topology:  rect
Neighborhood type:  gaussian
Learning rate type:  inverse
Initial learning rate parameter:  0.05 0.02
Initial radius of training area:  5 3
Average quantization error:  0.9505139
Average distortion measure:  25.37834 with error radius:  1
> som1$visual[1:10,]
   x y    qerror
1  0 0 0.6258783
2  0 0 0.8317910
3  0 0 0.7678554
4  0 0 0.7578742
5  0 0 0.7031048
6  0 0 0.6870237
7  0 0 0.7565792
8  0 0 0.6441827
9  0 0 0.7953218
10 0 0 0.6724060
```

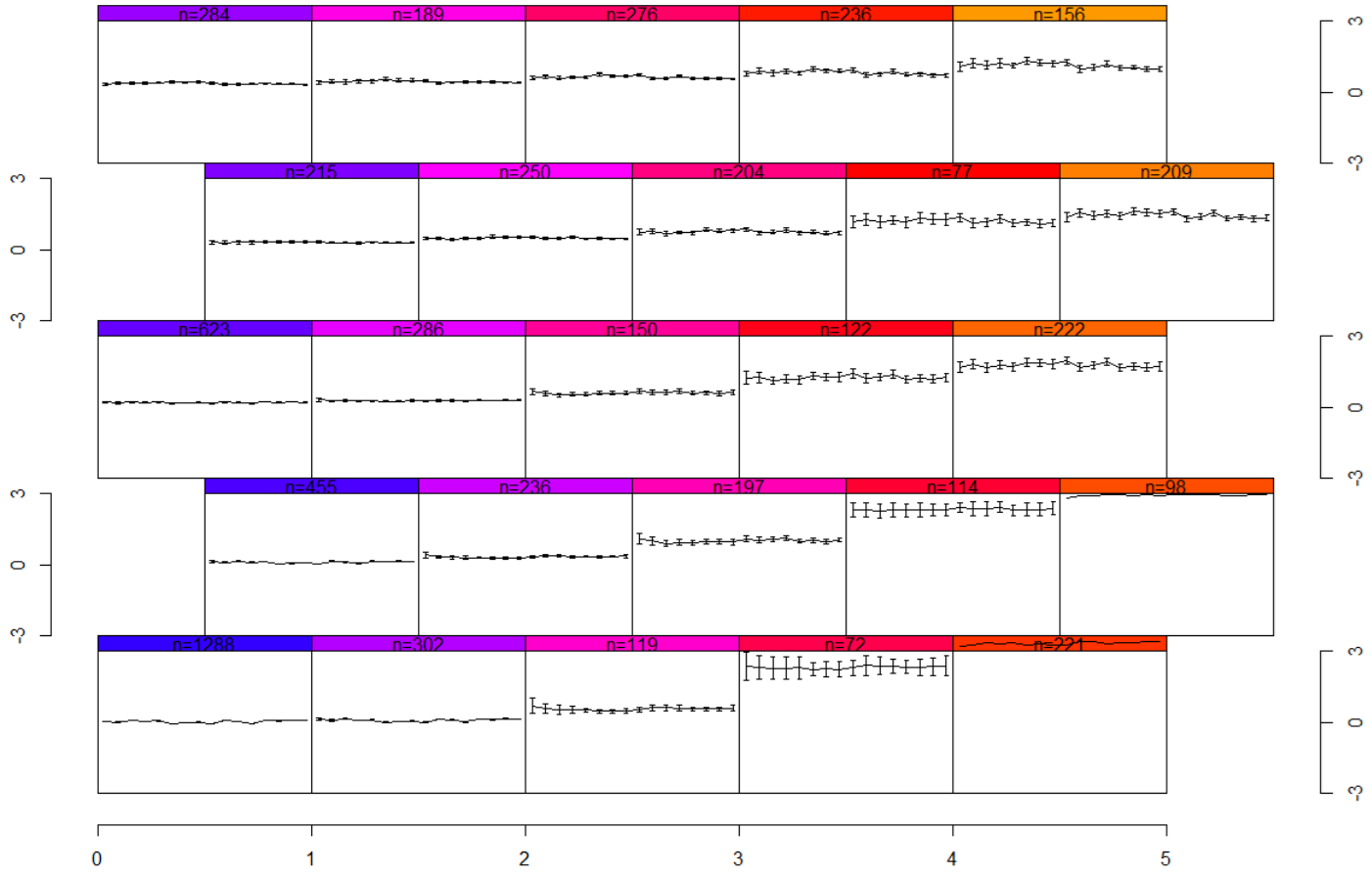
자기조직화지도: SOM

- Package som



자기조직화지도: SOM

- Package som



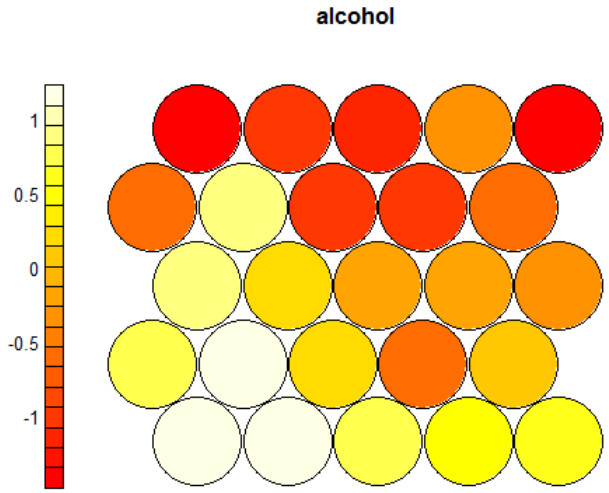
자기조직화지도: SOM

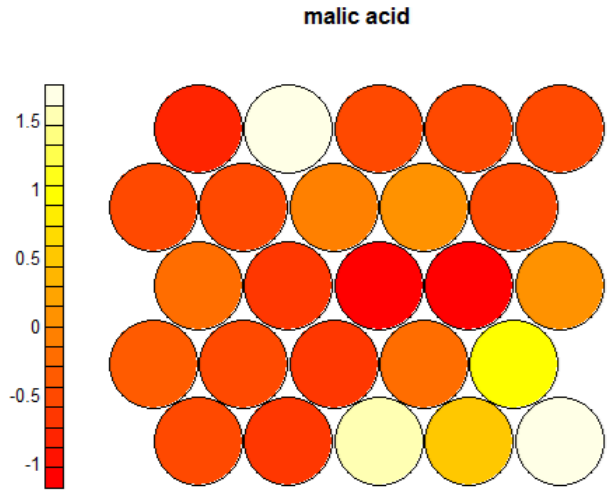
❖ Package kohonen

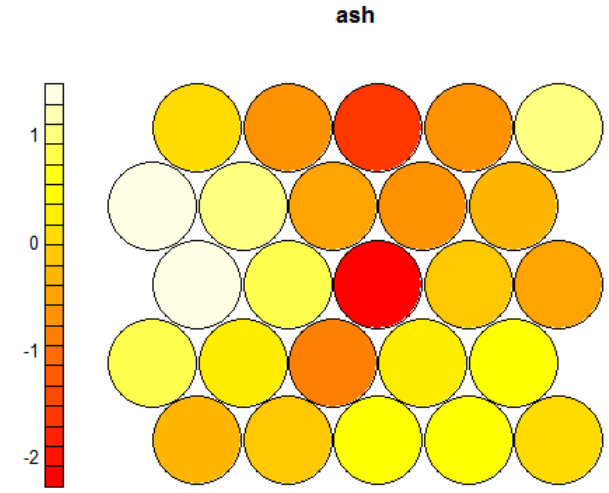
```
107 # kohonen package install
108 install.packages("kohonen", dependencies = TRUE)
109 detach("package:som", unload=TRUE)
110 library(kohonen)
111
112 data(wines)
113 trn = sample(nrow(wines), 120)
114 wines_trn <- scale(wines[trn,])
115 wines_tst <- scale(wines[-trn,], center = attr(wines_trn, "scaled:center"), scale = attr(wines_trn,"scaled:scale"))
116
117 som.wines <- som(wines_trn, grid = somgrid(5,5,"hexagonal"))
118 map(som.wines, wines_tst)
119 wine.cluster <- cutree(hclust(dist(som.wines$codes)), 3)
120
121 graphics.off()
122 par(mfrow = c(2, 2))
123 plot(som.wines, type = "property", property = som.wines$codes[,1], main = colnames(som.wines$codes)[1])
124 plot(som.wines, type = "property", property = som.wines$codes[,2], main = colnames(som.wines$codes)[2])
125 plot(som.wines, type = "property", property = som.wines$codes[,3], main = colnames(som.wines$codes)[3])
126 plot(som.wines, type = "property", property = som.wines$codes[,4], main = colnames(som.wines$codes)[4])
127 graphics.off()
128
129 par(mfrow = c(2, 2))
130 plot(som.wines, type = "quality")
131 plot(som.wines, type = "codes")
132 plot(som.wines, type = "changes")
133 dev.off()
134
135 som.pal <- c("#1f77b4", '#ff7f0e', '#2ca02c', '#d62728', '#9467bd', '#8c564b', '#e377c2')
136 plot(som.wines, type="mapping", bgcol = som.pal[wine.cluster], main = "Clusters")
137 add.cluster.boundaries(som.wines, wine.cluster)
```

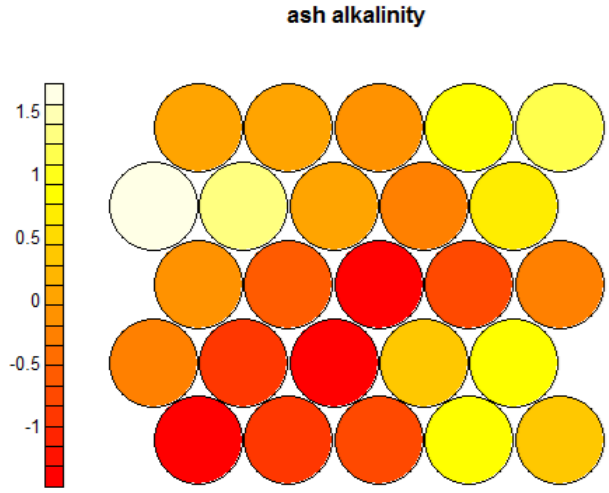
자기조직화지도: SOM

❖ Package kohonen



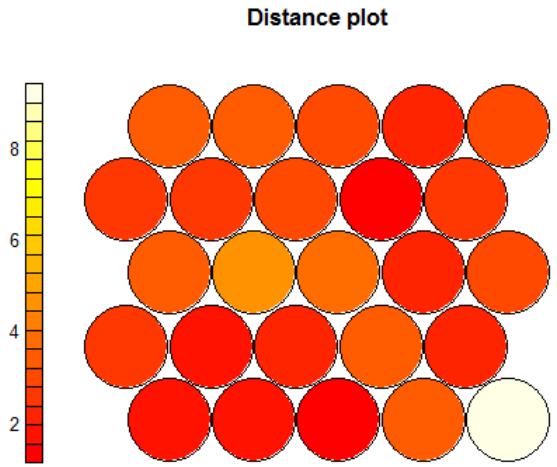


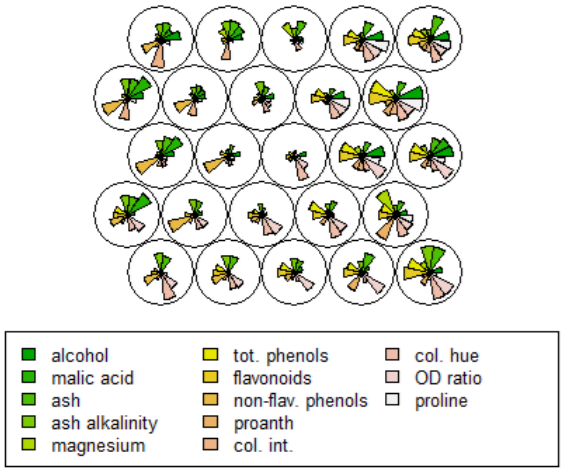


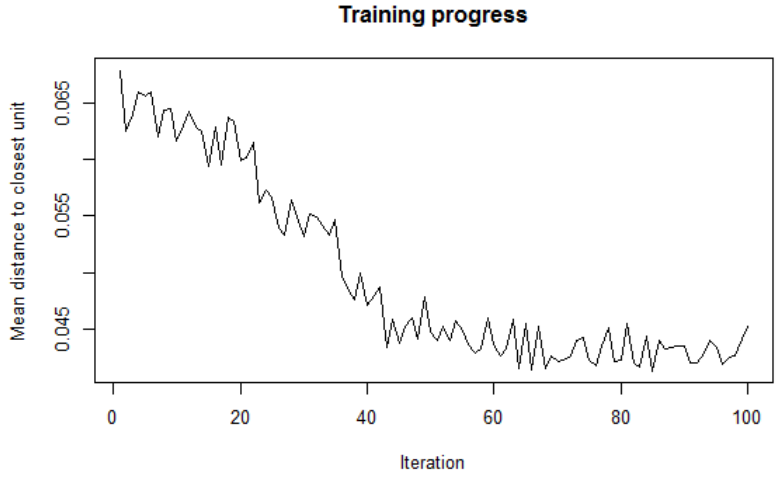


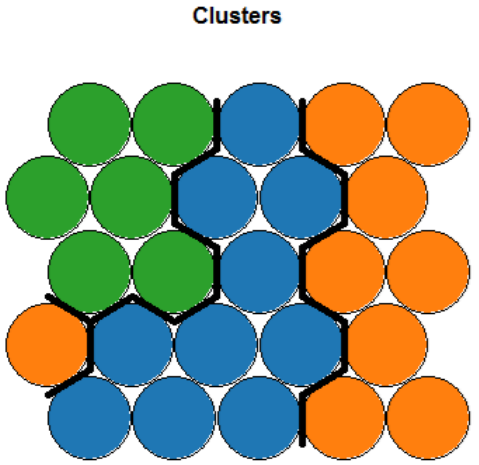
자기조직화지도: SOM

- Package kohonen









Q & A